

# On the Relationship Between Deterministic and Probabilistic Directed Graphical Models: from Bayesian Networks to Recursive Neural Networks

Pierre Baldi<sup>a,b,\*</sup> Michal Rosen-Zvi<sup>a,b</sup>

<sup>a</sup>*School of Information and Computer Science, University of California, Irvine CA  
92697-3425*

<sup>b</sup>*Institute for Genomics and Bioinformatics, University of California, Irvine CA  
92697-3425*

---

## Abstract

Machine learning methods that can handle variable-size structured data such as sequences and graphs include Bayesian networks (BNs) and Recursive Neural Networks (RNNs). In both classes of models, the data is modeled using a set of observed and hidden variables associated with the nodes of a directed acyclic graph. In BNs, the conditional relationships between parent and child variables are probabilistic whereas in RNNs they are deterministic and parameterized by neural networks. Here we study the formal relationship between both classes of models and show that when the source nodes variables are observed, RNNs can be viewed as limits, both in distribution and probability, of BNs with local conditional distributions that have vanishing covariance matrices and converge to delta functions. Conditions for uniform convergence and approximate bounds are also given together with an analysis of the behavior and exactness of Belief Propagation (BP) in “deterministic” BNs. Implications for the design of mixed architectures and the corresponding inference algorithms, as well as relationships to constraint networks, are briefly discussed.

*Key words:* Bayesian networks, belief propagation, recursive neural networks, recurrent neural networks, constraint networks, graphical models

---

---

\* Corresponding author.

*Email address:* [pfbaldi@ics.uci.edu](mailto:pfbaldi@ics.uci.edu) (Pierre Baldi).

*URL:* [www.ics.uci.edu/~pfbaldi](http://www.ics.uci.edu/~pfbaldi) (Pierre Baldi).

## 1 Introduction

Many problems in artificial intelligence, data mining, and machine learning involve variable-size structured data. By structured data we mean data that presents itself with an explicit data structure such as strings and sequences, trees, and directed or undirected graphs. Examples of structured data include: (1) text and documents in information retrieval; (2) DNA/RNA/protein sequences and evolutionary trees in bioinformatics; and (3) molecular structures in chemical informatics. To extract meaning, patterns, and regularities from these data requires computational methods that can not only handle structured data but also leverage structural information. Two classes of machine learning methods that have been applied to structured data are probabilistic graphical models [23, 16, 15, 12], such as Bayesian networks, and recursive neural networks [3, 13, 27, 17, 11, 21, 20, 4]. The purpose of this article is to analyze the mathematical relationship between these two approaches and, in particular, to show how a recursive neural network can be viewed as a limit of, or a fast approximation to, a sequences of Bayesian networks.

Bayesian networks (BNs) are probabilistic graphical models which rely on the global factorization of the joint probability distribution of a set of random variables into a product of local conditional probability distributions. More specifically, the random variables are associated with the nodes of a DAG (directed acyclic graph) and the local conditional distributions are the conditional distributions of a node variable, given the parent node variables. The global factorization is equivalent to a set of independence assumptions between the variables which generalizes the standard Markov independence assumptions for linear chains to more complex DAG structures. Technically speaking, a BN is defined on a fixed DAG that somehow reflects the structure of the data. In order to process data of variable size, we must use a dynamic Bayesian network where the basic underlying BN structure—also called a plate—is repeated multiple times, with a repetition number that depends on the data size and with parameters that are tied across the repetitions. For simplicity, in what follows, we use the term BNs in its broadest sense to include also dynamic BNs. Various kinds of Markov models, such as Hidden Markov Models (HMMs) and Bidirectional IOHMM (Input-Output HMMs) (Figure 1), are special cases of (dynamic) Bayesian networks which have been successfully applied to time series and sequential data ranging from speech to DNA and protein sequences [2]. Bayesian networks provide a flexible tool for dealing with structured data by capturing the structure of the data and of the inferences to be carried directly into the topology of the underlying DAG. In the case of large graphs and complex problems, however, the full probabilistic treatment, including information/belief propagation and learning, is often computationally challenging.

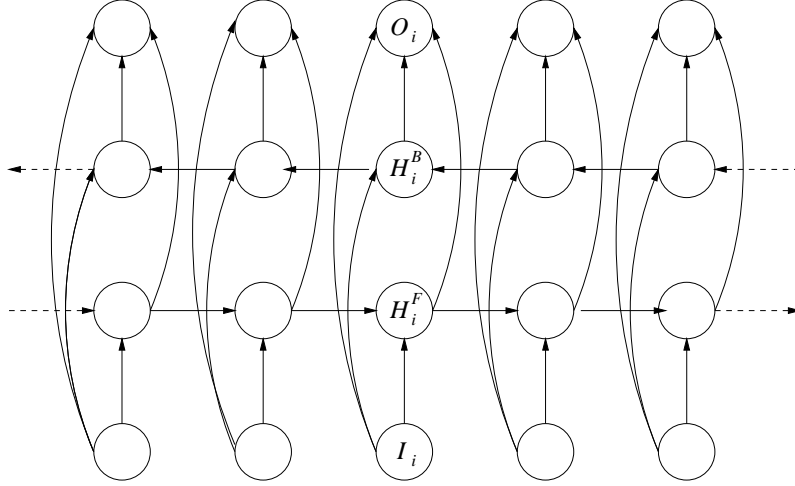


Fig. 1. DAG associated with input variables, output variables, and both forward and backward chains of hidden variables. As a Bayesian network, this is a bi-directional input-output HMM that has been used to model biological sequences.

Recursive neural networks provide an alternative to Bayesian networks for processing structured data. Recursive neural networks rely also on an underlying DAG but replace the probabilistic relationships between parents and child variables with a deterministic relationship parameterized by a neural network. In many applications the regular, translation-invariant, structure of the DAG allows reusing the same network at different locations in the graph—the so-called weight-sharing approach—leading to recurrent or recursive neural networks called DAG-RNNs [4].

As an example, consider the DAG of Figure 1 which has been used to model biological sequences. As a Bayesian networks this can be viewed as a bidirectional IOHMM. As a DAG-RNN model, however, the relationship between the variables can be modeled using three types of feed-forward neural networks to compute the output, forward, and backward variables respectively. One fairly general form of weight sharing is to assume stationarity for the output, forward, and backward networks, which finally leads to a 1D DAG-RNN architectures, implemented using three neural networks  $\mathcal{NN}_O$ ,  $\mathcal{NN}_F$ , and  $\mathcal{NN}_B$  in the form

$$\begin{aligned}
 O_i &= \mathcal{NN}_O(I_i, H_i^F, H_i^B) \\
 H_i^F &= \mathcal{NN}_F(I_i, H_{i-1}^F) \\
 H_i^B &= \mathcal{NN}_B(I_i, H_{i+1}^B)
 \end{aligned} \tag{1}$$

where  $i = 1, \dots, N$  and  $N$  is the length of the sequence being processed, as depicted in Figure 2. In this form, the output depends on the local input  $I_i$  at position  $i$ , the forward (upstream) hidden context  $H_i^F \in \mathbb{R}^n$  and the backward (downstream) hidden context  $H_i^B \in \mathbb{R}^m$ , with usually  $m = n$ . The

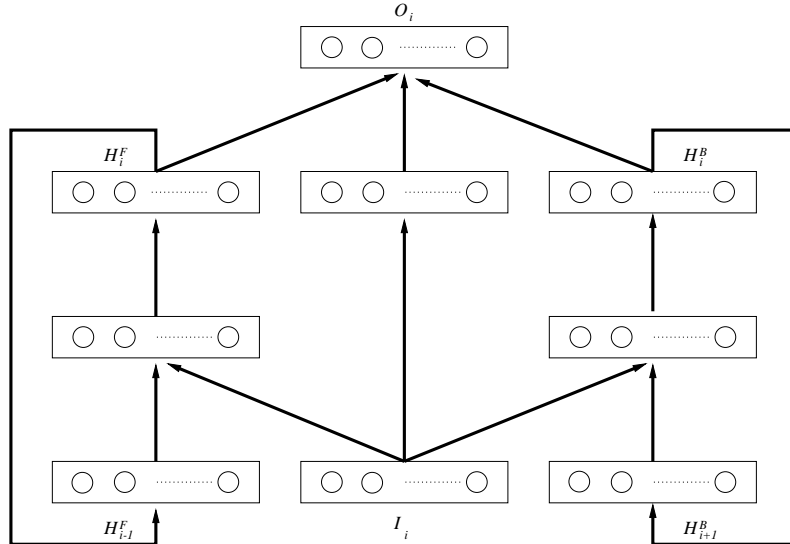


Fig. 2. A recursive neural network architecture associated with the DAG of Figure 1. At each position  $i$ , the output is computed by the same neural network which receives three input vectors: the local input vector  $I_i$ , and the left (forward)  $H_i^F$  and right (backward)  $H_i^B$  context vectors associated with two recurrent networks. These act as “wheels” that are rolled over the entire sequence up to position  $i$ . See [4] for additional details.

boundary conditions for  $H_i^F$  and  $H_i^B$  can be set to 0, i.e.  $H_0^F = H_{N+1}^B = 0$ . Alternatively these boundaries can also be treated as a learnable parameter. Intuitively, we can think of  $\mathcal{NN}_F$  and  $\mathcal{NN}_B$  in terms of two “wheels” that can be rolled along the sequence. For the prediction at position  $i$ , we roll the wheels in opposite directions starting from both ends and up to position  $i$ . Then we combine the wheel outputs at position  $i$  together with the input  $I_i$  to compute the output prediction  $O_i$  using  $\mathcal{NN}_O$ .

It should be clear that the deterministic relationship between parent and child variables can also be parameterized by other classes of functions and we shall refer to this general class of models as DAG-F models. It is essential to note that the DAG nature of the underlying graph allows unfolding of the model in time or space without introducing any cycles and therefore learning model parameters from examples can proceed using, for instance, gradient descent methods (backpropagation through time, space, or structure). The loss in semantic power resulting from the deterministic relationship in DAG-F models is compensated by the very fast deterministic feed-forward propagation of input evidence and therefore faster learning, which can be crucial in large-scale machine learning applications.

Deterministic relationships between parents and child variables in a DAG arise naturally also in constraint satisfaction networks [9] and as a mean to simplify and accelerate learning and inference in complex BN models. In [5], for instance, a Markovian BNs is constructed where the conditional distributions

of the hidden node variables are delta functions associated with the state of the parents. More generally, we define a deterministic Bayesian network (dBN) to be a BN where all the local conditional probability distributions are delta functions.

In this paper, we clarify the relationship between BNs on one hand and dBNs and DAG-F models on the other. In particular we show in which sense a dBN with its underlying DAG-F model can be viewed as a limit of a sequence of BNs when the local conditional distributions have vanishing covariance matrices.

## 2 Background and Notations

### 2.1 Directed Acyclic Graphs and Related Variables

Given a DAG  $G = (V, \vec{E})$  we always assume that its  $|V| = N$  nodes are labeled  $1, \dots, N$  in a topological order, i.e., the nodes are labeled with consecutive integers so that every arc is directed from a node with smaller label to a node with larger label. In what follows we do not distinguish the nodes and their labels, so that  $i < j$  implies that  $(j, i)$  is not an element of  $\vec{E}$ . A source node is a node with only outgoing edges and a sink nodes is a node with only incoming edges. Any DAG obviously has at least one source node and at least one sink node.  $\pi_i$  stands for the ordered list of parents of node  $i$ . If a node  $i$  has two parents  $j < j'$ , for example, then  $\pi_i = (j, j')$ .

The node hierarchy of a DAG ensures that the nodes can be partitioned into disjoint layers denoted  $K_0, K_1, \dots, K_{\max}$ . The layers are defined recursively letting  $K_0$  be the set of all source nodes.  $K_1$  is the set of all nodes in  $V - K_0$  that receive connections exclusively from nodes in  $K_0$ .  $K_k$  is the set of all nodes in  $V - \cup_{i=0}^{k-1} K_i$  that receive connections exclusively from nodes in  $K_0 \cup \dots \cup K_{k-1}$  and  $K_{\max}$  is the set that includes the sink nodes with the longest directed path from the source nodes, so that  $V = \cup_{i=0}^{\max} K_i$ . Note that the layers contain ascending lists of nodes in the sense that for all  $i \in K_k$  and  $j \in K_l$ , if  $k < l$  then  $i < j$ .

Real random vector variables or real vector values associated with the nodes of a DAG are denoted in the obvious way by  $X_i$  and  $x_i$  respectively, with  $x_i$  in  $\mathbb{R}^{n_i}$ . Similarly,  $x_K$  denote the ordered set of vectors associated with the ordered set  $K$ .

## 2.2 DAG-F Models

A DAG-F model (Figure 3) is a straightforward generalization of DAG-RNN defined by a labeled DAG as above, an integer  $n_i$  and corresponding vector variable  $X_i$  in  $\mathbb{R}^{n_i}$  for each node  $i = 1, \dots, N$ , and a set of real valued functions  $f_i$  associated with each node in  $V - K_0$ . In addition if  $\pi_i = i_1, \dots, i_{i_k}$  is the ordered list of parent variables of  $i$ , then the function  $f_i$  is a function from  $\mathbb{R}^{n_{i_1}} \times \dots \times \mathbb{R}^{n_{i_k}}$  to  $\mathbb{R}^{n_i}$ . A *consistent* set of vectors  $x_i$  for  $i = 1, \dots, N$  is such that for every  $i$  in  $V - K_0$  we have  $x_i = f_i(x_{\pi_i})$ . Thus a DAG-F is a graphical representation/decomposition of a real vector valued function. The input is described by the values that are entered at all the source nodes and the output is read out at the sink nodes for the corresponding consistent assignment of values which is trivially obtained by forward propagation, i.e. by computing the functions  $f_i$  layer by layer, starting with  $K_1$ . We denote this deterministic propagation by  $F$  so that, for any non source node  $i$  there is a deterministic function  $F_i$  such that  $x_i = F_i(x_{K_0})$ . The results in this paper are true both in the discrete and continuous case. In the continuous case, we will assume in general that the functions  $f_i$ , and hence also  $F_i$ , are continuous

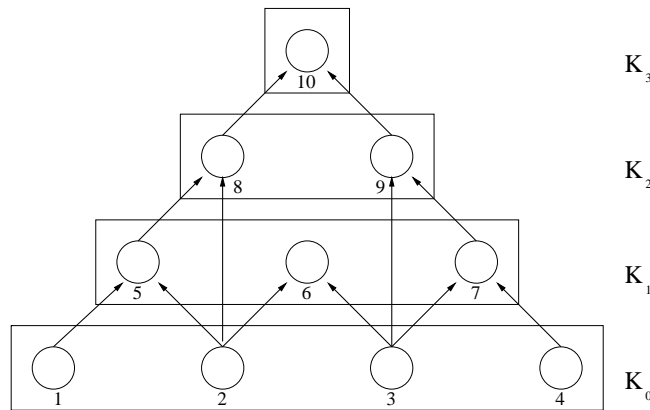


Fig. 3. DAG-F with 10 nodes with a consistent ordering and partitioned into 4 layers  $K_0, \dots, K_4$ . All source nodes are in  $K_0$ . Nodes 6 and 10 are the only sink nodes. If all the functions  $f_i$  correspond to addition and if the visible input is given by  $x_1 = x_2 = x_3 = x_4 = 1$  then, in a consistent assignment,  $x_5 = x_6 = x_7 = 2$ ,  $x_8 = x_9 = 3$ , and  $x_{10} = 6$ .

## 2.3 Bayesian Networks

A BN is defined by a DAG, a set of random variables associated with the vertices of the DAG, and a set of conditional distributions of each node variable given the parent node variables. The set of independence assumptions encoded by the graph implies the decomposition of the joint probability distribution into the product of all the local conditional distributions. If  $\pi_i = (i_1, \dots, i_{i_k})$  is

the ordered list of parent nodes of  $i$ , then the conditional probability density function,  $\rho_i(X_i|x_{\pi_i})$ , is a function  $\rho_i : \mathbb{R}^{n_i} \times \mathbb{R}^{n_{i_1}} \times \dots \times \mathbb{R}^{n_{i_k}} \rightarrow \mathbb{R}$  such that

$$P(X_i \in R) = \int_R \rho_i(x|x_{\pi_i}) dx \quad (2)$$

Here  $R$  defines a region in the  $n_i$  dimensional space. A complete description requires also giving the prior distribution of the variables associated with the source nodes. While we use a continuous notation here and in most of the article, it should be clear that the results are the same in the discrete case.

Given evidence in the form of the value assumed by some of the random variables, we can compute the posterior marginal distributions for any subset of the remaining variables by integrating out any residual variables. In most of the article, we will be concerned with the case where only source node variables may be observed, fully or partially, including the case where none is observed. An important special case that is particularly relevant in connection with DAG-F models is when *all* source nodes variables are observed, i.e. the full input case. In this case:

$$\rho(X_i|x_{K_0}) = \int \rho(X_{V-K_0}|x_{K_0}) \prod_{j \in V-K_0-i} dX_j \quad (3)$$

Here  $\rho(X_i|x_{K_0})$  denotes the posterior marginal probability distribution of  $X_i$  given the observed source nodes. Likewise,  $\rho(X_{V-K_0}|x_{K_0})$  denotes the joint probability distribution over the unobserved random variables,  $X_{V-K_0}$  conditioned on the known values  $x_{K_0}$ . This joint probability distribution is factorized into the product of the local distributions according to the underlying graph,  $\rho(X_{V-K_0}|x_{K_0}) = \prod_{j \in V-K_0} \rho_i(X_j|x_{\pi_j})$ . Note that we use the subscript  $i$  for the local conditional distribution that define the BN but we omit it for the local posterior marginal distribution. A similar relation holds for the posterior marginal of clusters of node variables. For any cluster  $X_S$

$$\rho(X_S|x_{K_0}) = \int \rho(X_{V-K_0}|x_{K_0}) \prod_{j \in V-K_0-S} dX_j \quad (4)$$

#### 2.4 Deterministic Bayesian Networks

By a deterministic Bayesian network we mean a Bayesian network where all the local conditional probability functions are Kronecker or Dirac delta functions, in the discrete and continuous case respectively, so that

$$\rho_i(X_i|x_{\pi_i}) = \delta(X_i - f_i(x_{\pi_i})) \quad (5)$$

for some function  $f_i$ . It should be clear that there is a one-to-one correspondence between DAG-F and dBN models via the functions  $f_i$ . The DAG-F associated with a dBN, however, is deprived of the probabilistic semantics present in the corresponding dBN. In particular, in a DAG-F evidence can only be entered in the source nodes—this is not the case for the corresponding dBN in general. While in a dBN all nodes have deterministic behavior, it is of course possible to consider mixed cases where only a strict subset of the node variables of a BN is associated with delta functions. This is the case, for instance, for the model described in [5] in generative mode (during learning all the nodes are deterministic). Further examples are given in the Conclusion.

## 2.5 Sigma Bayesian Networks

Finally, we introduce the notion of  $\sigma$ BN associated with a DAG-F or DBN, by considering families of BNs with the same underlying DAG-F and node variables, where the local conditional probability functions are almost deterministic, i.e. have vanishing small entropy. This can be achieved by having a vanishingly small covariance matrix controlled by a vanishing parameter  $\sigma$ . The particular form of the distribution is not important for our results, but, to fix the ideas, the reader may consider continuous systems with Gaussian conditional distributions of the form:

$$\rho_i(X_i|x_{\pi_i}) = \mathcal{N}(X_i; f_i(x_{\pi_i}), \sigma^2 I) \quad (6)$$

with mean  $f_i(x_{\pi_i})$  and covariance matrix  $\sigma^2 I$ , where  $I$  is the  $n_i \times n_i$  identity matrix. The more general requirement, in the continuous case, is that the sequence of conditional distributions be continuous around the limit point. In fact, the covariance matrix need not be diagonal. Any covariance matrix will do as long as the variance and covariance terms converge to 0 to yield delta function behavior. Likewise the conditional probabilities do not have to be Gaussian. Other hill-shaped distributions that converge to delta functions will also work. Depending on the situation, one could use for instance rectangles of width  $2\sigma$  and height  $1/2\sigma$ , or Dirichlet distributions in the case of variables associated with multinomial distributions. A Dirichlet distribution on the probability vector  $P = (p_1, \dots, p_n)$ , with parameters  $\alpha$  and  $Q = (q_1, \dots, q_n)$ , has the form

$$D_{\alpha Q}(P) = \frac{\Gamma(\alpha)}{\prod_i \Gamma(\alpha q_i)} \prod_{i=1}^n p_i^{\alpha q_i - 1} = \prod_{i=1}^n \frac{p_i^{\alpha q_i - 1}}{Z(i)} \quad (7)$$

with  $\alpha, p_i, q_i \geq 0$  and  $\sum p_i = \sum q_i = 1$ . For such a Dirichlet distribution,  $E(p_i) = q_i$ ,  $Var(p_i) = q_i(1 - q_i)/(\alpha + 1)$ , and  $Cov(p_i p_j) = -q_i q_j / (\alpha + 1)$ . Thus

letting the control parameter be  $\sigma = 1/\alpha$  ensures that both variances and covariances converge to 0 with  $\sigma$ .

## 2.6 Belief Propagation in Bayesian Networks with Source Node Evidence Only

One of the most common inference algorithms used for deriving approximate marginals in BNs is the Belief propagation (BP) algorithm [24]. In general, Pearl's BP algorithm for directed graphs includes messages from parents to children and from children to parents. However, in the case of BNs with partial or full evidence associated with source nodes only, BP becomes a purely feedforward algorithm. More precisely, the backward messages from children to parents do not contain any relevant information and can be ignored so that the posterior marginal can be approximated recursively, from source to sink nodes, by

$$\tilde{\rho}(X_i) = \int \rho_i(X_i|X_{\pi_i}) \prod_{j \in \pi_i} [\tilde{\rho}_j(X_j) dX_j] \quad (8)$$

Here  $\tilde{\rho}(X_i)$  is the message in the BP approximation and it is easy to show that each of these messages is also a probability distribution intended to approximate the posterior marginal of  $X_i$ . This result was proved for discrete random variables in [8] and for distributions from the exponential family in [25]. It can easily be generalized along the same lines to any distribution.

As an illustrative example, we consider the dBN in Figure 3 where all conditional distributions are defined by Equation 6 with vanishing  $\sigma$ , all functions  $f_i$  correspond to addition and the observed input is given by  $x_1 = x_2 = x_3 = x_4 = 1$ . The messages of all nodes in  $K_0$  are Dirac delta functions,  $\tilde{\rho}(x_i) = \delta(x_i - 1)$ , since these values are observed. When  $\sigma = 0$  and using Equation 8 hierarchically, one finds immediately that for all nodes in layer 1  $\tilde{\rho}(x_i) = \delta(x_i - 2)$ , for all nodes in layer 2  $\tilde{\rho}(x_i) = \delta(x_i - 3)$ . Finally, using messages from node 8 and 9 the message at node 10 is found to be  $\tilde{\rho}(x_{10}) = \delta(x_{10} - 6)$ . Later we show that these values are exact in the dBN case, and can be used as good approximations when  $\sigma$  is non-zero but sufficiently small.

## 2.7 Constraint Networks

The BP algorithm has also been shown to belong to a broader family of algorithms used in areas ranging from information coding to constraint networks [26, 1, 19]. A dBN and the associated DAG-F can also be viewed as constraint

networks. Relationships between constraint networks and BNs, and between the arc-consistency algorithm of constraint networks and BP are explored further in [10], and [18].

## 2.8 Convergence Problems

In this paper, for a fixed DAG-F and the fixed associated dBN, we study the convergence properties of the corresponding  $\sigma$ BNs to the dBN as  $\sigma \rightarrow 0$  when only the source nodes are fully or partially observed. That is in which sense can we say that “ $\lim_{\sigma \rightarrow 0} \sigma\text{BN} = \text{dBN}(\text{DAG-F})$ ”? More specifically, we address two different problems. First, in section 3, we study the convergence of the posterior marginals of the  $\sigma$ BN to the posterior marginals of the dBN. Then, in section 4, we study the convergence of the approximate posterior marginals produced by BP in  $\sigma$ BNs with source evidence only to the corresponding dBN posterior marginals, as  $\sigma \rightarrow 0$ . In both cases, we analyze both weak convergence, i.e. in distribution, and strong convergence, i.e. in probability, as well as conditions for uniform convergence. As a byproduct, we also show that Belief Propagation, in a dBN where only the source nodes are fully or partially observed, is an exact (and purely feedforward) algorithm.

The exactness of BP in dBNs with fully observed source nodes is intuitively clear within the general framework provided by [23] where BP is shown to be exact on trees or, equivalently, it is exact in cases where all parents of each child are independent, i.e., for any child  $i$ , and  $\{j, j'\} \in \pi_i$ ,  $X_j \perp X_{j'}$  and thus  $\rho_{\pi_i}(X_{\pi_i}) = \prod_{j \in \pi_i} \rho_j(X_j)$ . The marginal of each particular node in a dBN is a delta function over the single possible value that that node is likely to take. Similarly, the marginal of the cluster of parents is provided by a product of delta functions that define the single set of values that these nodes are likely to take. Such factorization in a dBN justifies the BP assumptions which leads to beliefs that are exact posterior marginals.

## 3 Convergence of Posterior Marginals

We first study the convergence properties of posterior marginals of single nodes. The generalization to posterior marginals of bigger clusters is straightforward. We deal with the case where all the input variables are observed and then show how the same ideas can be applied when some or all the input variables are unobserved.

### 3.1 Convergence in Distribution

*Theorem 3.1 (Convergence in Distribution):* Let  $x_{K_0}$  denote a complete set of evidence at the source nodes of a  $\sigma$ BN with an underlying DAG-F. Then for any node  $i$  in  $G$ ,

$$\lim_{\sigma \rightarrow 0} \rho(X_i | x_{K_0}) = \delta(X_i - F_i(x_{K_0})) \quad (9)$$

in other words all the local marginal distributions converge in distribution to delta functions centered at the consistent deterministic values provided by the underlying DAG-F.

*Proof:* The result is obvious for the source nodes. We then consider the case where  $i$  is the sink with the largest index. Because of the BN factorization, we have

$$\rho(X_{V-K_0} | x_{K_0}) = \prod_{i \in V-K_0} \rho_i(X_i | x_{\pi_j}) \quad (10)$$

Applying the definition of marginalization in Equation 3 yields

$$\lim_{\sigma \rightarrow 0} \rho(X_i | x_{K_0}) = \lim_{\sigma \rightarrow 0} \int \prod_{j \in V-K_0-K_1} \rho_i(X_j | x_{\pi_j}) \prod_{j \in K_1} \rho_i(X_j | x_{\pi_i}) \prod_{j \in V-K_0-i} dX_j \quad (11)$$

The product over the first layer nodes is distinguished from the product over all other nodes to emphasize that the parents of that layer are observed. Interchanging the limit and integral operators and using the fact that the local conditional distributions in a  $\sigma$ BN converge to delta functions yields

$$\lim_{\sigma \rightarrow 0} \rho(X_i | x_{K_0}) = \int \prod_{j \in V-K_0-K_1} \delta(X_j - f_j(x_{\pi_j})) \prod_{j \in K_1} \delta(X_j - F_j(x_{K_0})) \prod_{j \in V-K_0-i} dX_j \quad (12)$$

where  $\delta(X_i - f_i)$  is the  $n_i$ -dimensional Dirac delta function. The limit and the integration operators can be interchanged because all the densities are assumed to be continuous, at least around the limit points. Thus we can apply the well known theorem that if  $g$  is continuous around  $x$  and  $f_n \rightarrow \delta(x)$  then  $\lim_{n \rightarrow \infty} \int f_n g = g(x)$ . Performing the integration recursively, in ascending order, gives the result

$$\lim_{\sigma \rightarrow 0} \rho(X_i | x_{K_0}) = \delta(X_i - F_i(x_{K_0})) \quad (13)$$

The calculation is very similar for all the other nodes and relies on the facts that in Equation 12 the integral  $\int dX_j \delta(X_j - f_j(X)) = 1$  no matter what the value of  $f_j(X)$  is (provided it exists and is finite, of course) and furthermore for any continuous function  $f$   $\int f(x) \delta(y - x) dx = f(y)$ . This theorem can easily be extended to clusters  $X_S$  of variables, in the form

$$\lim_{\sigma \rightarrow 0} \rho(X_S | x_{K_0}) = \prod_{j \in S} \delta(X_j - F_j(x_{K_0})) \quad (14)$$

An alternative proof of this result can be obtained by an application of Slutsky's theorem. Slutsky's theorem states that: If  $X_n$  converges in distribution to  $X$  and  $Y_n$  converges in distribution (or in probability) to  $c$ , a constant, then  $X_n + Y_n$  converges in distribution to  $X + c$ . More generally, if  $f(x, y)$  is continuous then  $f(X_n, Y_n)$  converges in distribution to  $f(X, c)$ . Slutsky's theorem applies immediately to the case above with both  $X_n$  and  $Y_n$  converging to constants.

The same result is first obtained in the case of unobserved discrete bounded variables in the source nodes by considering each input configuration separately with its corresponding probability. The posterior marginals then become mixtures of delta functions,

$$\lim_{\sigma \rightarrow 0} \rho(X_i) = \sum_{x_{K_0}} \prod_{j \in K_0} p(x_j) \delta(X_i - F_i(x_{K_0})) \quad (15)$$

where  $p(x_j)$  is the given probability that the  $j$  random variable (in the source node) equals  $x_j$ . In the case of unbounded variables or of continuous variables, the same result is obtained by considering compact supports and taking the limit,

$$\lim_{\sigma \rightarrow 0} \rho(X_i) = \prod_{j \in K_0} \rho(x_j) \delta(X_i - F_i(x_{K_0})) \quad (16)$$

Empirically, this amounts to sampling the input variables according to their distribution and, for each sample and for each node, computing the posterior marginal as a delta function centered on the corresponding value provided by the underlying DAG-F.

In fact, an even stronger form of convergence holds.

### 3.2 Convergence in Probability

*Theorem 3.2 (Convergence in Probability):* Let  $x_{K_0}$  denote a complete set of evidence at the source nodes of a  $\sigma$ BN with an underlying DAG-F. Then for

any node  $i$  in  $G$ , and for any  $\epsilon$

$$\lim_{\sigma \rightarrow 0} P(|(X_i|x_{K_0}) - F_i(x_{K_0})| > \epsilon) = 0 \quad (17)$$

Here  $X_i|x_{K_0}$  is a random variable distributed according to the posterior distribution,  $\rho(X_i|x_{K_0})$ .

In other words, the marginal random variables converge in probability to the corresponding consistent constant values. This results from the general fact that if a random variable converges in distribution to a constant, then it converges in probability to that constant [7]. In the general case where some of the input variables are not observed, the result above can be immediately extended in the case of discrete finite input variables, by taking an OR over all possible input configurations.

We immediately get *uniform* convergence across the finite set of nodes in  $G$  and across the finite set of examples by minimizing the value of  $\sigma$  in the corresponding convergence inequalities. By taking limits over the set of examples, the result remains true over an infinite set of examples, as long as the set is compact (i.e. closed and bounded) and the functions  $f_i$ , hence  $F_i$ , are continuous (hence bounded).

### 3.3 Uniform Convergence

*Theorem 3.3 (Uniform Convergence in Probability):* Consider a  $\sigma$ BN with an underlying DAG-F. For every  $\epsilon > 0$  and every  $\alpha > 0$ , there is an integer  $m$  such that if  $\sigma < 1/m$  then for every node  $i$  in  $G$  and any complete evidence  $x_{K_0}$  in a compact set  $C$ ,

$$P(|(X_i|x_{K_0}) - F_i(x_{K_0})| > \epsilon) < \alpha \quad (18)$$

provided the functions  $f_i$  are continuous. In other words, there is convergence in probability uniformly across all the nodes in a BN and across all the evidence inputs in a compact set. This also implies fast convergence of the associated Gibbs sampler in the sense that smaller and smaller sample sizes are required to derive mean values. Similar results are obtained in the discrete case.

Thus for a given covariance matrix  $\sigma I$  and any value of  $\epsilon$  there exists a minimal value of  $\alpha$  such that the inequality above holds. The minimal value of  $\alpha$  depends in addition on the shape of the functions  $f_i$ , the dimensionality of the random variables, and the specific DAG. In the Appendix, we derive estimates for  $\alpha$  under simple assumptions.

## 4 Exactness and Convergence of Belief Propagation

In this section, we turn to the relationship between the BP beliefs (posterior marginals) in  $\sigma$ BNs and in dBNs. We use the convergence results of the previous section to prove exactness of BP in dBNs and derive bounds on the error of the BP approximation in  $\sigma$ BNs.

### 4.1 Convergence in Distribution

*Theorem 4.1 (Convergence in Distribution of BP):* Let  $x_{K_0}$  denote a complete set of evidence at the source nodes of a  $\sigma$ BN with an underlying DAG-F. Then for any node  $i$  in  $G$ ,

$$\lim_{\sigma \rightarrow 0} \tilde{\rho}(X_i | x_{K_0}) = \delta(X_i - F_i(x_{K_0})) \quad (19)$$

in the discrete case, or in the continuous case provided the functions  $f_i$  (hence  $F_i$ ) are continuous.

*Proof:* The source nodes are observed hence their BP messages towards the first layer are all delta functions. We now proceed by induction layer by layer. We consider a node  $i$  in a given layer and assume that the property is true for all the nodes in all the previous layers. Using the definition of BP given by Equation 8 we have

$$\lim_{\sigma \rightarrow 0} \tilde{\rho}(X_i) = \lim_{\sigma \rightarrow 0} \int \rho_i(X_i | X_{\pi_i}) \prod_{j \in \pi_i} \tilde{\rho}_j(X_j) dX_j \quad (20)$$

$$= \int \lim_{\sigma \rightarrow 0} \rho_i(X_i | X_{\pi_i}) \prod_{j \in \pi_i} [\delta(X_j - F_j(x_{K_0}))] dX_j \quad (21)$$

$$= \lim_{\sigma \rightarrow 0} \rho_i(X_i; F_i(x_{K_0}), \sigma^2 I) \quad (22)$$

$$= \delta(X_i - F_i(x_{K_0})) \quad (23)$$

Equation 20 uses continuity to interchange the integral with the limit operator in the continuous case. The delta functions result from the induction hypothesis. In other words, all the local posterior marginal distributions derived by BP in  $\sigma$ BN converge in distribution to delta functions centered at the consistent deterministic values of the underlying DAG-F. An alternative proof of this result can also be obtained using Slutsky's theorem.

The same convergence-in-distribution result was proved in Section 3 for the exact posterior marginals. Together, these two convergence results, prove that

BP is exact in dBNs.

#### 4.2 Exactness of BP

*Theorem 4.2 (Exactness of BP-derived Posterior Marginals in dBNs):* Let  $x_{K_0}$  denote a complete set of evidence at the source nodes of a dBN with an underlying DAG- F. Let  $\tilde{\rho}$  denote the approximated posterior marginals derived by BP. Then for any node  $i$

$$\tilde{\rho}(X_i|x_{K_0}) = \rho(X_i|x_{K_0}) \tag{24}$$

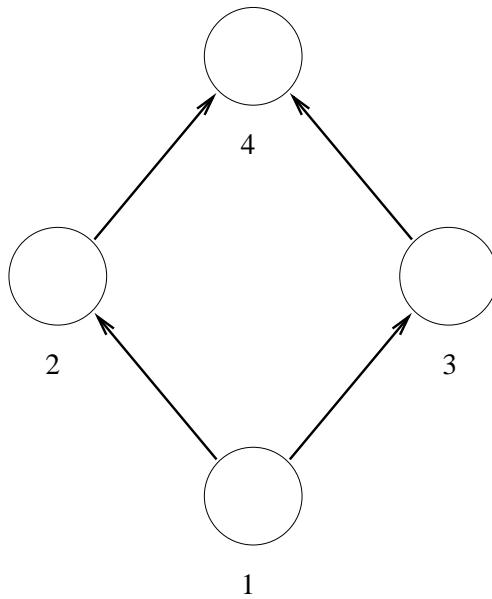


Fig. 4. DAG-F with 4 nodes and one loop.  $x_1$  is the source variable and the functions are  $f_2 = f_3 = Id = x_1$ ,  $f_4 = x_2 \cdot x_3$ .

Clearly both the convergence in distribution of the BP beliefs as  $\sigma \rightarrow 0$  and the exactness of the BP beliefs in dBNs can be used to derive exact marginals in the case where the source nodes are only partially observed, or not observed at all. In a procedure that is different from the standard BP updates, one can derive estimates of the posterior marginal distributions by combining the BP beliefs obtained for each possible fully observed setting of the source nodes. This procedure, that we call decomposition of BP marginals, is based on the well-known cutset conditioning method (see [23, 22] for details) for inferring posterior marginals. When the domain of the source nodes is discrete, we simply run BP on each possible realization of the source nodes and decompose the posterior marginal probability of a sink node, or any other node, accordingly using the distribution of the source nodes.

As a simple illustration of decomposition of BP marginals, consider the DAG in Figure 4, with a single loop, a single source node, and a single sink node, and functions  $f_2 = f_3 = x_1$  and  $f_4 = x_2 \cdot x_3$ . Let us assume that the source variable can take two values, 1 and -1, with a uniform probability so that the prior can be written as  $\rho(X_1) = \frac{1}{2} [\delta(X_1 - 1) + \delta(X_1 + 1)]$ . Then for  $X_1 = 1$  in a dBN, application of BP yields the sink distribution  $\rho(X_4|X_1 = 1) = \delta(X_4 - 1)$ . In this case also for  $X_1 = -1$  one obtains  $\rho(X_4|X_1 = -1) = \delta(X_4 - 1)$ . The probability distribution of the sink for unobserved source nodes is obtained simply by combining both results in the form  $\rho(X_4) = \int \frac{1}{2} [\delta(X_1 - 1) + \delta(X_1 + 1)] \rho(X_4|X_1) dX_1 = \delta(X_4 - 1)$ , which is the exact result for this network. In this case the cutset contains only the source node.

In the case where the source variables have an infinite domain, one can apply the decomposition of BP marginals by running BP symbolically on fixed values of the source nodes and then compose the resulting marginals using the prior distribution on the source variables. In the example above, if  $X_1$  assumes continuous values and the prior is changed to  $\rho(X_1) = \mathcal{N}(X_1; 0, 1)$  and the set of conditional distributions remains the same (dBN), then each value  $x_1$  of the source yields the value  $x_1^2$  for the sink variable by BP. Therefore  $\rho(X_4) = \int \mathcal{N}(X_1; 0, 1) \delta(X_4 - X_1^2) dX_1 = \frac{\exp(-X_4/2)}{2\sqrt{2\pi X_4}}$ . Note that due to the correlations between the parents of  $X_4$ , in both examples above the standard BP marginals are *not* exact, while the composition of BP posterior marginal distributions does provide exact posterior marginal distributions.

Moreover, the exactness of BP in dBNs applies not only to posterior marginals of single node variables but also to posterior marginals of larger clusters of variables as well. Indeed, the BP algorithm of Equation 8 provides posterior marginals for single nodes and these can be used for deriving marginals of larger clusters. For instance, the estimation of the posterior marginal a cluster associated with a family is  $\tilde{\rho}(X_i, X_{\pi_i}) = \rho_i(X_i|\pi_i) \prod_{j \in \pi_i} \tilde{\rho}_j(X_j)$ . Therefore, it is sufficient to use the convergence properties of the BP posterior marginals of single nodes. The convergence in distribution of single node beliefs implies that

$$\lim_{\sigma \rightarrow 0} \tilde{\rho}(X_i, X_{\pi_i}) = \rho_i(X_i|\pi_i) \prod_{j \in \pi_i} \delta(X_j - F_j(x_{K_0})) \quad (25)$$

which, in turns, implies exactness in dBNs:  $\tilde{\rho}(X_i, X_{\pi_i}) = \rho(X_i, X_{\pi_i})$ . Using again the example associated with Figure 4, one can easily see that the BP approximation for observed source node, say  $X_1 = -1$ , and dBN provides the marginal estimation  $\rho(X_2, X_3, X_4) = \delta(X_2 + 1)\delta(X_3 + 1)\delta(X_4 - 1)$  which is the exact marginal of the cluster.

### 4.3 Convergence in Probability of BP

*Theorem 4.3 (Convergence in Probability of BP Posterior Marginals):* Let  $x_{K_0}$  denote a complete set of evidence at the source nodes of a  $\sigma$ BN with an underlying DAG-F. Let  $\tilde{\rho}$  denote the approximated posterior marginals derived by BP. Then for any node  $i$  in  $G$ , and for any  $\epsilon > 0$

$$\lim_{\sigma \rightarrow 0} \tilde{\rho}(|(X_i|x_{K_0}) - F_i(x_{K_0})| > \epsilon) = 0 \quad (26)$$

To prove this one has to use Equation 8 and follow the lines of the proof of convergence in probability in Section 3. Using the same reasoning as in the previous Section, it is easy to prove also uniform convergence.

### 4.4 Uniform Convergence

*Theorem 4.4 (Uniform Converges in Probability of BP Posterior Marginals):* Consider a  $\sigma$ BN with an underlying DAG-F. Let  $\tilde{\rho}$  denote the approximated posterior marginals derived by BP. For every  $\epsilon > 0$  and every  $\alpha > 0$ , there is an integer  $m$  such that if  $\sigma < 1/m$  then for every node  $i$  in  $G$  and any complete evidence  $x_{K_0}$  in a compact set  $C$ ,

$$\tilde{\rho}(|X_i|x_{K_0} - F_i(x_{K_0})| > \epsilon) < \alpha \quad (27)$$

provided the functions  $f_i$  are continuous.

In other words, in a  $\sigma$ BN with small  $\sigma$ , BP provides posterior marginals that are close to the underlying DAG-F results.

By combining the facts that dBN (DAF-F-derived) posterior marginals are close to both the exact and the BP-derived posterior marginals in the corresponding  $\sigma$ BN, we can see that the BP-derived posterior marginals are also close to the exact posterior marginals in  $\sigma$ BNs. Indeed, in a  $\sigma$ BN with hard evidence in the source nodes, let  $X_i$  denote a sample obtained from the true posterior marginal distribution and  $Y_i$  a sample obtained from the posterior marginal distribution estimated by BP. We know that for any  $\epsilon$  and any  $\alpha$ , for  $\sigma$  small enough, we must have

$$P(|X_i - F_i(x_{K_0})| > \epsilon) < \alpha \quad \text{and} \quad P(|Y_i - F_i(x_{K_0})| > \epsilon) < \alpha \quad (28)$$

These bounds imply a relation between the samples drawn from the true distribution and from the BP approximation

$$P(|X_i - Y_i| > 2\epsilon) \leq P\{|X_i - F_i(x_{K_0})| > \epsilon \quad \text{or} \quad |Y_i - F_i(x_{K_0})| > \epsilon\} \quad (29)$$

which yields immediately

$$P(|X_i - Y_i| > 2\epsilon) < 2\alpha \quad (30)$$

Note that several distances, such as the Kullback Liebler divergence and the  $L^p$  ( $1 \leq p \leq \infty$ ) distances between  $\rho$  and  $\tilde{\rho}$  are not necessarily bounded. In the Appendix, we show how to derive rough estimates for  $\epsilon$  and  $\alpha$  using standard linearization and Gaussian approximations.

## 5 Conclusion

In summary, deterministic relationships between parents and child variables in a directed acyclic graph (DAG) arise naturally in constraint satisfaction networks, in recursive neural networks associated with DAGs (DAG-RNNs), and as a mean to simplify and accelerate learning and inference in probabilistic graphical models, such as Bayesian networks. A deterministic Bayesian network (dBN) is a Bayesian network where all the conditional probability distributions of a node variable given its parent variables are Kronecker or Dirac delta functions. A sigma Bayesian network ( $\sigma$ BN) is a corresponding family of Bayesian networks, with the same underlying DAG and node variables, where the local conditional distributions have covariance matrices that converge to 0 together with a control parameter  $\sigma$  (e.g. Gaussians with vanishing covariance matrices). Here we have shown that when the source nodes are observed fully or partially, the posterior marginals of a  $\sigma$ BN converge to the posterior marginals of the corresponding dBN both in distribution and in probability, as  $\sigma$  approaches 0. In addition, the approximate posterior marginals computed by the Belief Propagation algorithm in the  $\sigma$ BN also converge to the posterior marginals in the corresponding dBN, both in distribution and in probability. This implies that Belief Propagation is an exact feedforward algorithm in dBNs with source node evidence only. Conditions for uniform convergence and approximate bounds are also given.

In DAG-F models, the evidence at the source nodes may include what are traditionally called inputs but also other boundary nodes, especially when stationarity (or “weight-sharing”) assumptions are used for some of the functions  $f_i$ . For instance, in an input-output HMM architecture [6], in addition to the inputs the source nodes include also the boundary nodes of the hidden

Markov chain. In DAG-RNNs these inputs are typically set to 0 vectors, occasionally they can also be viewed as additional parameters to be learnt from the data.

Although internal propagation inside a DAG-F is deterministic, the overall model itself can remain probabilistic. This is the case, for instance, with DAG-RNNs used in classification where the values computed in the output layer correspond to class probabilities, computed by logistic or normalized exponential neural units. In this case, the range of some of the variables  $x_i$  can be restricted to classification probability values and, strictly speaking, we can use Dirichlet distributions rather than Gaussians to define the conditional probability distributions of the corresponding nodes, given their parents. Thus, in spite of their deterministic variables, DAG-Fs and dBNs can remain full-fledged probabilistic models. They can be viewed as self-standing models, or as limiting cases of BNs, where the introduction of deterministic units speeds up inference and may render complex models tractable.

We have analyzed the convergence of  $\sigma$ BNs to dBNs and the underlying DAG-F as the parameter  $\sigma$  goes to 0 and the properties of BP in  $\sigma$ BNs, and dBNs. We have shown that BP is exact in dBNs and derived error bound for the BP marginals in  $\sigma$ BNs. Thus if in a BN the conditional dependency relations can be reasonably modeled or approximated by deterministic relations, then DAG-F propagation in the corresponding dBN can be used to derive posterior marginals that are exact for the dBN and reasonable approximations for the original posterior marginals. From a practical standpoint, our results are not meant to suggest that a DAG-F or dBNs should be replaced by taking the limit of some  $\sigma$ BN but rather the opposite. In some situations it may be possible to replace, simplify, or approximate a portion of a BN using dBNs or DAG-F models to speed up belief propagation and learning. In particular, we can apply these results to BNs that are combinations of dBNs and trees, since BP can provide exact posterior marginal distributions for each one of these components. Here we shall give two simple examples to illustrate the ideas.

Consider first, the case of a BN where we can partition the nodes of the underlying DAG into a loop cutset and its complement. If the nodes in the cutset are deterministic (observed or dBN), then BP provides exact posterior marginals in the cutset and its complement, thus on the entire BN. The special case of BNs with binary random variables, where the loop cutset consist of a single node with a  $\sigma$ BN structure, is studied in [8].

In the second example, consider a BN such that the graph associated with layers  $K_1$  to  $K_l$  is a tree with non-deterministic random variables, and the graph associated with the layers from  $K_l$  to  $K_{max}$  contains loops but is a dBN. In this case BP provides the exact posterior marginals for all nodes in  $K_1$  to  $K_l$ , due to the tree structure. One can view  $K_l$  as the source nodes of

the dBN (or  $\sigma$ BN) with known marginals for all the source nodes. Thus all Theorems above apply for the posterior marginals of the nodes in  $K_l$  to  $K_{max}$  and in particular one can apply the decomposition of BP marginals and again obtain exact marginals for the entire BN.

Finally, several directions of theoretical research remain open. For instance, a better understanding of dBNs with evidence in the sink nodes only would be of interest and relevant to a better understanding of the relationship between BNs, dBN/DAG-Fs, and constraint satisfaction networks.

## Acknowledgement

The work of PB is in part supported by a Laurel Wilkening Faculty Innovation award, an NIH grant, and a Sun Microsystems award. MRZ is supported by an NSF grant. We would like to thank Rina Dechter, David van Dyk, and Max Welling for useful discussions.

## A Appendix: Bounds

In this appendix, we will use superscripts to keep track of the layers and will assume, for simplicity, that all variables are one dimensional. Consider a  $\sigma$ BN where the local conditional distributions are given by

$$X_i^l | x_{\pi_i}^{l-1} = f_i^l(x_{\pi_i}^{l-1}) + \alpha_i^l \quad (\text{A.1})$$

where  $\alpha_i^l$  is a random variable with mean zero and variance equal (or bounded) by  $\sigma^2$  and the alpha variables of two different nodes are supposed to be independent. We will also assume that the functions  $f_i^l$  are continuously differentiable for all  $i$  and all  $l$  and, when necessary, we will assume that the variables  $\alpha$  are normally distributed. The goal is to show by induction that for all  $i$  and all  $l$ , to a first degree of approximation

$$X_i^l | x_{K_0} \approx F_i^l(x_{K_0}) + \beta_i^l \quad (\text{A.2})$$

where the  $\beta$  random variables have mean zero and vanishing variances  $b_i^l$  as  $\sigma \rightarrow 0$ . We will also show that the variables  $\beta_i^l$  and  $\beta_j^l$  have vanishing covariances  $c_{ij}^l$  within the same layer. We let  $B^l = \max_i b_i^l$ ,  $B = \max_l B^l$ ,  $C^l = \max_{i,j} c_{ij}^l$ , and  $C = \max_l C^l$ . Let also  $D^l = \max(B^l, C^l)$ .

We have shown that the variables  $X$  can be approximated using BP. To show this, we approximate the variables  $X$  by BP using simple feedforward propagation. This hierarchical approach is related to but different from the derivation of the Extended Kalman Filter equations [14]. In the first layer, we have

$$X_i^1|x_{K_0} = f_i^1(x_{K_0}) + \alpha_i^1 = F_i^1(x_{K_0}) + \alpha_i^1 \quad (\text{A.3})$$

so the relationship is true with  $\beta_i^1 = \alpha_i^1$  and  $B^1 = \sigma^2$ ,  $C^1 = 0$ , and  $D^1 = \sigma^2$ . Suppose that the relationship in Equation A.3 is true up to layer  $l-1$ . Consider a variable  $X_i^l$  in layer  $l$ , with parent variables  $X_{i_1}^{l-1}, \dots, X_{i_k}^{l-1}$ . Then

$$X_i^l|x_{K_0} \approx f_i^l(X_{i_1}^{l-1}, \dots, X_{i_k}^{l-1}) + \alpha_i^l \quad (\text{A.4})$$

$$\approx f_i^l(F_{i_1}^{l-1}(x_{K_0}) + \beta_{i_1}^{l-1}, \dots, F_{i_k}^{l-1}(x_{K_0}) + \beta_{i_k}^{l-1}) + \alpha_i^l \quad (\text{A.5})$$

By taking a first order Taylor approximation using the continuous differentiability of the functions  $f_i^l$ , this finally yields

$$X_i^l|x_{K_0} \approx F_i^l(x_{K_0}) + \sum_{j=1}^k \frac{\partial f_i^l}{\partial x_{i_j}} \beta_{i_j}^{l-1} + \alpha_i^l \quad (\text{A.6})$$

so that Equation A.2 is satisfied with

$$\beta_i^l = \sum_{j=1}^k \frac{\partial f_i^l}{\partial x_{i_j}} \beta_{i_j}^{l-1} + \alpha_i^l \quad (\text{A.7})$$

Clearly, by linearity,  $E(\beta_i^l) = 0$ . For the variance term, we have

$$b_i^l = E(\beta_i^l \beta_i^l) = \sum_{j=1}^k \left( \frac{\partial f_i^l}{\partial x_{i_j}} \right)^2 b_{i_j}^{l-1} + \sum_{j=1}^k \sum_{m=1}^k \frac{\partial f_i^l}{\partial x_{i_j}} \frac{\partial f_i^l}{\partial x_{i_m}} c_{i_j i_m}^{l-1} + \sigma^2 \quad (\text{A.8})$$

with  $j \neq m$  in the double sum. Let  $P^l$  be the largest fan in, or number of parents, for a node in layer  $l$  and  $P = \max_l P^l$ . Let  $s_i^l = \max_j |\partial f_i^l / \partial x_{i_j}|$ ,  $S^l = \max_i s_i^l$ , and  $S = \max_l S^l$ . Then

$$b_i^l \leq P^l (s_i^l)^2 B^{l-1} + \frac{P^l(P^l - 1)}{2} (s_i^l)^2 C^{l-1} + \sigma^2 \quad (\text{A.9})$$

so that

$$B^l \leq P^l (S^l)^2 B^{l-1} + \frac{P^l(P^l - 1)}{2} (S^l)^2 C^{l-1} + \sigma^2 \leq (S^l)^2 D^{l-1} P^l \frac{P^l + 1}{2} + \sigma^2 \quad (\text{A.10})$$

For the covariance term,

$$c_{ij}^l = E(\beta_i^l \beta_j^l) = E\left(\sum_{u=1}^k \sum_{v=1}^r \frac{\partial f_i^l}{\partial x_{i_u}} \beta_{i_u}^{l-1} \frac{\partial f_j^l}{\partial x_{j_v}} \beta_{j_v}^{l-1}\right) \quad (\text{A.11})$$

where  $r$  is the number of parents of node  $j$ . Separation of variance and covariance terms depends on the degree of overlap between the parents of  $i$  and  $j$ . Assume that  $o_{ij}^l$  is the number of parents in common. Then

$$c_{ij}^l \leq s_i^l s_j^l [o_{ij}^l B^{l-1} + (rk - o_{ij}^l) C^{l-1}] \leq (S^l)^2 [P^l B^{l-1} + (P^l)^2 C^{l-1}] \quad (\text{A.12})$$

so that

$$C^l \leq P^l (S^l)^2 D^{l-1} [P^l + 1] \quad (\text{A.13})$$

By combining Equations A.10 and A.13, when  $\sigma \leq 1$  we get

$$D^l \leq P(P+1)S^2 D^{l-1} + \sigma^2 \quad (\text{A.14})$$

with  $S^1 = \sigma^2$ . Thus a non particularly subtle bound gives

$$D \leq ([P(P+1)S^2]^{L-1} + 1) \sigma^2 \quad (\text{A.15})$$

Thus  $D$  which bounds all the variances and covariances of the variables  $\beta$  can be made as small as needed by reducing  $\sigma^2$ .

From this it is easy to find estimates for the bounds for the theorems on the convergence in probability. For instance, since  $X_i^l \approx F_i^l(x_{K_0}) + \beta_i^l$  and  $\beta_i^l$  has small variance  $b_i^l$ , we can use Thebycheff inequality to write:

$$P(|X_i^l - F_i^l(x_{K_0})| > \epsilon) \leq \frac{b_i^l}{\epsilon^2} \quad (\text{A.16})$$

If the variables  $\beta_i^l$  are Gaussian, then more precise inequalities can be derived of course

$$P(|X_i^l - F_i^l(x_{K_0})| > \epsilon) = 2 \int_{\frac{\epsilon}{\sqrt{2\pi b_i^l}}}^{+\infty} \frac{1}{\sqrt{2\pi b_i^l}} e^{-t^2/2(b_i^l)^2} dt \quad (\text{A.17})$$

which yields the exponentially decreasing upper bound of

$$\frac{2}{\sqrt{2\pi b_i^l}} e^{-\epsilon^2/2(b_i^l)^2} \quad (\text{A.18})$$

for small values of  $b_i^l$ .

## References

- [1] S. M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, 2000.
- [2] P. Baldi and S. Brunak. *Bioinformatics: the machine learning approach*. MIT Press, Cambridge, MA, 2001. Second edition.
- [3] P. Baldi and Y. Chauvin. Hybrid modeling, HMM/NN architectures, and protein applications. *Neural Computation*, 8(7):1541–1565, 1996.
- [4] P. Baldi and G. Pollastri. The principled design of large-scale recursive neural network architectures—DAG-RNNs and the protein structure prediction problem. *Journal of Machine Learning Research*, 4:575–602, 2003.
- [5] D. Barber. Dynamic Bayesian networks with deterministic latent tables. In *Advances in Neural Information Processing Systems 12*, 2000.
- [6] Y. Bengio and P. Frasconi. An input-output HMM architecture. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 7. Morgan Kaufmann, San Francisco, CA, 1995. (IEEE).
- [7] P. Billingsley. *Probability and Measure*. Wiley, New York, NY, 1995. Third Edition.
- [8] B. Bozhena and R. Dechter. The epsilon-cutset effect in bayesian networks. Technical report, School of Information and Computer Science, University of California, Irvine, 2001.
- [9] R. Dechter. *Constraint Processing*. Morgan Kauffman, 2003.
- [10] R. Dechter and J. Pearl. Directed constraint networks: A relational framework for causal modeling. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1991.
- [11] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE Transactions on Neural Networks*, 9(5):768–786, 1998.
- [12] B. J. Frey. *Graphical Models for Machine Learning and Digital Communication*. MIT Press, 1998.
- [13] C. Goller and A. Kuchler. Learning task-dependent distributed structure-representations by backpropagation through structure. *IEEE International Conference on Neural Networks*, pages 347–352, 1996.
- [14] S. Haykin, editor. *Kalman Filtering and Neural Networks*. John Wiley & Sons, 2001.
- [15] D. Heckerman. A tutorial on learning with Bayesian networks. In M.I. Jordan, editor, *Learning in Graphical Models*. Kluwer, Dordrecht, 1998.
- [16] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, UK, 1996.

- [17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [18] R. Mateescu and R. Decther. A simple insight into properties of iterative belief propagation. In *Uncertainty in Artificial Intelligence*, 2003.
- [19] R. J. McEliece and M. Yildirim. Belief propagation on partially ordered sets. In D. Gilliam and J. Rosenthal, editors, *Mathematical Systems Theory in Biology, Communications, and Finance*. IMA, University of Minnesota, 2002.
- [20] A. Micheli, A. Sperduti, A. Starita, and A. M. Bianucci. A novel approach to QSPR/QSAR based on neural networks for structures. In H. Cartwright and L. M. Sztandera, editors, *Soft Computing Approaches in Chemistry*, pages 265–296. Springer Verlag, Heidelberg, Germany, 2003.
- [21] A. Micheli, A. Sperduti, A. Starita, and A. M. Bianucci. Analysis of the internal representations developed by neural networks for structures applied to quantitative structure-activity relationship studies of benzodiazepines. *J. Chem. Inf. Comput. Sci.*, 41:202–218, 2001.
- [22] J Pearl. Fusion, propagation, and structuring in belief networks. *Artif. Intell.*, 29(3):241–288, 1986.
- [23] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, San Mateo, CA, 1988.
- [24] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [25] M. Rosen-Zvi and M. I. Jordan. Approximate inference and the DLR equations. Technical report, Computer Science Division, University of California, Berkeley, 2003.
- [26] P. P. Shenoy and G. Shafer. Axioms for probability and belief-function propagation. In R. D. Shachter, T. S. Levitt, J. F. Lemmer, and L. Kanal, editors, *Uncertainty in Artificial Intelligence*, volume 4. North-Holland, Amsterdam, 1990.
- [27] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

names]label.