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

Pierre Baldi [a,b,*] Michal Rosen-Zvi [a,b]

[a] *School of Information and Computer Sciences, University of California, Irvine CA 92697-3425*

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

Preprint submitted to Elsevier Science *17 June 2005*

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: (a) 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 [14, 10] such as Bayesian networks, and recursive neural networks [1, 9, 16, 11, 8, 12, 2]. 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. 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 of BNs, including information/belief propagation and learning, can be computationaly challenging.

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 [2].

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 fast deterministic propagation of forward input evidence and backwork output errors, which is crucial in large-scale machine learning applications.

Deterministic relationships between parents and child variables in a DAG arise naturally also in constraint satisfaction networks [7] and as a mean to simplify and accelerate learning and inference in complex BN models. In [4], 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. Technical details, including bounds and proofs of theorems, as well as material on constraint networks are omitted for brevity but can be found in the corresponding technical report [3] downloadable from `www.ics.uci.edu/~pfbaldi/publications.htm`.

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. π_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 $K_0, K_1, ..., K_{\text{max}}$. 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 $V - \cup_{i=0}^{i=k-1} K_i$ that receive connections exclusively from nodes in $K_0 \cup \ldots \cup K_{k-1}$ and K_{max} is the set that includes the sink nodes with the longest directed path from the source nodes, so that $V = \cup_{i=0}^{\text{max}} K_i$. 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 $I\!R^{n_i}$. 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 1) 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 $I\!R^{n_i}$ for each node $i = 1, \ldots, N$, and a set of real valued functions f_i associated with each node in $V - K_0$. In addition if $\pi_i = i_1, \ldots i_{i_k}$ is the ordered list of parent variables of i, then the function f_i is a function from $I\!R^{n_{i_1}} \times \ldots I\!R^{n_{i_{i_k}}}$ to $I\!R^{n_i}$. A *consistent* set of vectors x_i for $i = 1, \ldots, N$ 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

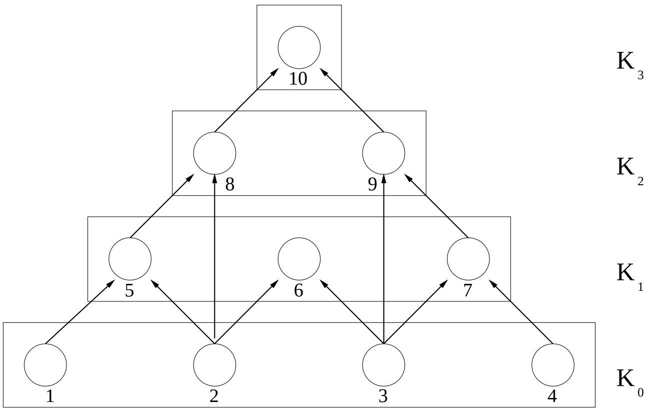

Fig. 1. DAG-F with 10 nodes with a consistent ordering and partitioned into 4 layers $K_0, \ldots, K_4$. All source nodes are in K_0. Nodes 6 and 10 are the only sink nodes. If all the functions f_i correspond to addition and if the visible input is given by $x_1 = x_2 = x_3 = x_4 = 1$ then, in a consistent 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 $\pi_i = (i_1, \ldots i_{i_k})$ 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 $\rho_i : I\!\!R^{n_i} \times I\!\!R^{n_{i_1}} \times \ldots I\!\!R^{n_{i_{i_k}}} \rightarrow I\!\!R$ such that $P\,(X_i \in R) = \int_R \rho_i(x|x_{\pi_i})dx$. 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: $\rho\,(X_i|x_{K_0}) = \int \rho(X_{V-K_0}|x_{K_0}) \prod_{j \in V-K_0-i} dX_j$. 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.

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 $\rho_i(X_i|x_{\pi_i}) = \delta(X_i - f_i(x_{\pi_i}))$, 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 [4] 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 σBN 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 σ. 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:

$$\rho_i\left(X_i|x_{\pi_i}\right) = \mathcal{N}\left(X_i; f_i(x_{\pi_i}), \sigma^2 I\right) \tag{1}$$

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 2σ and height $1/2\sigma$, or Dirichlet distributions in the case of variables associated with multinomial distributions.

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 [14]. 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

$$\tilde{\rho}(X_i) = \int \rho_i(X_i|X_{\pi_i}) \prod_{j\in\pi_i} [\tilde{\rho}_j(X_j)dX_j] \qquad (2)$$

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 [6] and for distributions from the exponential family in [15]. It can easily be generalized along the same lines to any distribution.

As an illustrative example, we consider the dBN in Figure 1 where all conditional distributions are defined by Equation 1 with vanishing σ, 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 2 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 σ is non-zero but sufficiently small.

In this paper, for a fixed DAG-F and the fixed associated dBN, we study the convergence properties of the corresponding σBNs to the dBN as $\sigma \to 0$ when only the source nodes are fully or partially observed. That is in which sense can we say that "$\lim_{\sigma\to 0}\sigma\text{BN} = \text{dBN(DAG-F)}$"? More specifically, we address two different problems. First, in section 3, we study the convergence of the posterior marginals of the σBN to the posterior marginals of the dBN. Then, in section 4, we study the convergence of the approximate posterior marginals produced by BP in σBNs with source evidence only to the corresponding dBN posterior marginals, as $\sigma \to 0$. 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 feeforward) algorithm.

3 Convergence of Posterior Marginals in Distribution

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.

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

$$\lim_{\sigma\to 0}\rho(X_i|x_{K_0}) = \delta(X_i - F_i(x_{K_0})) \tag{3}$$

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.

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,

$$\lim_{\sigma\to 0}\rho(X_i) = \sum_{x_{K_0}}\prod_{j\in K_0} p(x_j)\delta\left(X_i - F_i(x_{K_0})\right) \tag{4}$$

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,

$$\lim_{\sigma \to 0} \rho(X_i) = \prod_{j \in K_0} \rho(x_j) \delta \left(X_i - F_i(x_{K_0}) \right) \tag{5}$$

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.

Theorem 3.2 (Convergence in Probability): Let x_{K_0} denote a complete set of evidence at the source nodes of a σBN with an underlying DAG-F. Then for any node i in G, and for any ϵ

$$\lim_{\sigma \to 0} P(|(X_i|x_{K_0}) - F_i(x_{K_0})| > \epsilon) = 0 \tag{6}$$

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 [5]. 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 σ 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).

Theorem 3.3 (Uniform Convergence in Probability): Consider a σBN 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,

$$P(|(X_i|x_{K_0}) - F_i(x_{K_0})| > \epsilon) < \alpha \tag{7}$$

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.

4 Exactness and Convergence of Belief Propagation

In this section, we turn to the relationship between the BP beliefs (posterior marginals) in σBNs and in dBNs and use the convergence results of the previous section to prove exactness of BP in dBNs.

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

$$\lim_{\sigma \to 0} \tilde{\rho}(X_i | x_{K_0}) = \delta(X_i - F_i(x_{K_0})) \tag{8}$$

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

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.

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

$$\tilde{\rho}(X_i | x_{K_0}) = \rho(X_i | x_{K_0}) \tag{9}$$

Clearly both the convergence in distribution of the BP beliefs as $\sigma \to 0$ 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 [14, 13] 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.

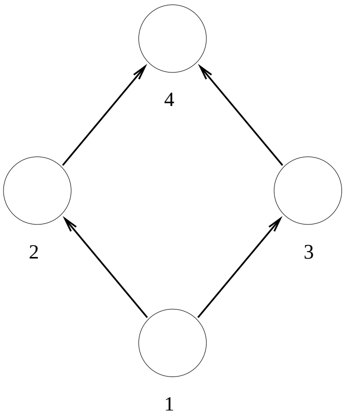

Fig. 2. DAG-F with 4 nodes and one loop. x_1 is the source variable and the functions are $f_2 = f_3 = Id = x_1$, $f_4 = x_2 \cdot x_3$.

As a simple illustration of decomposition of BP marginals, consider the DAG in Figure 2, 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 asssume that the source variable can take two values, 1 and -1, with a uniform probability so that the prior can be written as $\rho(X_1) = \frac{1}{2}\left[\delta(X_1 - 1) + \delta(X_1 + 1)\right]$. 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 $\rho(X_4) = \int \frac{1}{2}\left[\delta(X_1 - 1) + \delta(X_1 + 1)\right]\rho(X_4|X_1)dX_1 = \delta(X_4 - 1)$, 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 addition, the exactness of BP in dBNs applies not only to posterior marginals of single node variables, but also to posterior marginals of clusters of variables. For instance, in the dBN associated with Figure 2 with source evidence $X_1 = -1$, BP 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.

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 σBN 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 $\epsilon > 0$

$$\lim_{\sigma \to 0} \tilde{\rho}(|(X_i|x_{K_0}) - F_i(x_{K_0})| > \epsilon) = 0 \tag{10}$$

Theorem 4.4 (Uniform Converges in Probability of BP Posterior Marginals): Consider a σBN 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,

$$\tilde{\rho}(|X_i|x_{K_0} - F_i(x_{K_0})| > \epsilon) < \alpha \tag{11}$$

provided the functions f_i are continuous.

In other words, in a σBN with small σ, 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 σBN, we can see that the BP-derived posterior marginals are also close to the exact posterior marginals in σBNs.

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 (σBN) 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 σ (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 σBN converge to the posterior marginals of the corresponding dBN both in distribution and in probability, as σ approaches 0. In addition, the approximate posterior marginals computed by the Belief Propagation algorithm in the σBN 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.

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 of the data. 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 σBNs to dBNs and the underlying DAG-F as the parameter σ goes to 0 and the properties of BP in σBNs, and dBNs. We have shown that BP is exact in dBNs and derived error bound for the BP marginals in σBNs. 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 σBN 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 only 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 σBN structure, is studied in [6]. 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 σBN) 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.

Acknowledgement

Work supported by a Laurel Wilkening Faculty Innovation award, a Sun Microsystems award, and NSF and NIH grants to PB. We would like to thank R. Dechter, D. van Dyk, and M. Welling for discussions.

References

[1] P. Baldi and Y. Chauvin. Hybrid modeling, HMM/NN architectures, and protein applications. *Neural Computation*, 8(7):1541–1565, 1996.

[2] 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.

[3] P. Baldi and M. Rosen-Zvi. On the relationship between deterministic and probabilistic directed graphical models: from Bayesian networks to recursive neural networks and back. *Technical Report*, 2005. Department of Computer Science, University of California, Irvine.

[4] D. Barber. Dynamic Bayesian networks with determinsitic latent tables. In *Advances in Neural Information Processing Systems 12*, 2000.

[5] P. Billingsley. *Probability and Measure.* Wiley, New York, NY, 1995. Third Edition.

[6] 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.

[7] R. Dechter. *Constraint Processing.* Morgan Kauffman, 2003.

[8] 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.

[9] 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.

[10] D. Heckerman. A tutorial on learning with Bayesian networks. In M.I. Jordan, editor, *Learning in Graphical Models.* Kluwer, Dordrecht, 1998.

[11] 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.

[12] 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.

[13] J Pearl. Fusion, propagation, and structuring in belief networks. *Artif. Intell.*, 29(3):241–288, 1986.

[14] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[15] M. Rosen-Zvi and M. I. Jordan. Approximate inference and the DLR equations. Technical report, Computer Science Division, University of California, Berkeley, 2003.

[16] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.