Implementation of Continuous Bayesian Networks Using Sums of Weighted Gaussians

Eric Driver and Darryl Morrell
Telecommunications Research Center and
Department of Electrical Engineering
Arizona State University
Tempe, AZ 85287

Abstract

Bayesian networks provide a method of representing conditional independence between random variables and computing the probability distributions associated with these random variables. In this paper, we extend Bayesian network structures to compute probability density functions for continuous random variables. We make this extension by approximating prior and conditional densities using sums of weighted Gaussian distributions and then finding the propagation rules for updating the densities in terms of these weights. We present a simple example that illustrates the Bayesian network for continuous variables; this example shows the effect of the network structure and approximation errors on the computation of densities for variables in the network.

1 INTRODUCTION

Bayesian networks provide a method of representing conditional independence relationships between random variables and of computing the probability distributions associated with these random variables. Bayesian networks were originally developed for discrete-valued random variables; there is an increasing interest in extending this approach to continuous-valued random variables. Previous work on networks of continuous-valued variables has required that the variables have Gaussian density functions and that the relationships between these variables be linear (Pearl 1988, Shachter 1989). Theoretical issues involving the representation of relationships between random variables using a network (directed or undirected graph) consisting of both continuous and discrete random variables, including tests for conditional independence, are addressed in (Lauritzen 1989, Lauritzen 1990); an approach to updating conditional probabilities in which a single Gaussian is used to approximate a mixture of Gaussian densities is given in (Spiegelhalter 1990). In this paper, we extend Bayesian networks to include random variables with arbitrary distributions. We make this extension by approximating prior and conditional densities using sums of weighted Gaussian distributions; to our knowledge, this is the first time this approximation technique has been used in Bayesian networks. Using this technique, we find propagation rules for updating the densities of the network variables; information propagates through the network in the form of messages consisting of weight, mean, and variance updates.

2 PRELIMINARIES

We implement Bayesian networks for continuous variables by approximating both prior and conditional density functions by sums of weighted Gaussian densities. A density so approximated is represented in terms of a set of weights, a set of means, and a set of variances. Computation of these approximations is a hard problem which we discuss in Section 5.

Figure 1 shows a fragment of a Bayesian network. For our implementation, we assume that the network is singly connected (i.e. at most one path connects any two variables) and we assume that X is related to its
Implementation of Continuous Bayesian Networks Using Sums of Weighted Gaussians

Let \( g(x) \) denote the evidence in the subnetwork above \( X \) and let \( e_X^\pm \) denote the evidence in the subnetwork below \( X \). Also, let \( e_j^\pm(i = 1, 2, \ldots, n) \) denote the evidence coming to \( X \) via node \( U_i \), and let \( e_j^\pm(j = 1, 2, \ldots, m) \) denote the evidence coming to \( X \) via node \( Y_j \). We assume \( X \) has received messages \( \pi_X(u_i) \) from its parents; each message is a sum of weighted Gaussians:

\[
\pi_X(u_i) \stackrel{def}{=} \sum_{k=1}^{M_i} \alpha_i^k N(u_i; (\sigma^i_{\pi,k})^2, \mu^i_{\pi,k}),
\]

where \( M_i \) is the number of Gaussian densities in the representation for \( \pi_X(u_i) \), \( \{\alpha_i^k\} \) is a set of real valued weights, \( \{\sigma^i_{\pi,k}\} \) is a set of real valued variances, and \( \{\mu^i_{\pi,k}\} \) is a set of real valued means. Similarly, we assume \( X \) has received messages \( \lambda_{Y,j}(x)(j = 1, 2, \ldots, m) \) from its children:

\[
\lambda_{Y,j}(x) \stackrel{def}{=} f(x_j|e_j^\pm) = \begin{cases} 
\sum_{l_j=1}^{P_j} \beta_j^{l_j} N(x_j; (\sigma_{\lambda,j})^2, \mu_{\lambda,j}) & \text{if } e_j^\pm \neq 0 \\
1 & \text{if } e_j^\pm = 0
\end{cases}
\]

where \( P_j \) is the number of Gaussian densities in the representation for \( \lambda_{Y,j}(x) \), \( \{\beta_j^{l_j}\} \) is a set of real valued weights, \( \{(\sigma_{\lambda,j})^2\} \) is a set of real valued variances, and \( \{\mu_{\lambda,j}\} \) is a set of real valued means.

The belief function for \( X \) is computed as follows:

\[
BEL(x) \stackrel{def}{=} f(x|e) = \int f(x|e^+_X, e^-_X) f(e^-_X|x) \, d(e^-_X) = \alpha \pi(x),
\]

where \( \alpha = \frac{1}{f(e^+_X|e^-_X)} \) is a normalization constant.

We compute \( \pi(x) \) as follows:

\[
\pi(x) \stackrel{def}{=} f(x|e^+_X) = \int_{u_1}^{\cdots} \int_{u_n}^{\cdots} f(x|e^+_X, u_1, \ldots, u_n) \cdot f(u_1, \ldots, u_n|e^+_X) \, du_1 \cdots du_n
\]

\[
= \int_{u_1}^{\cdots} \int_{u_n}^{\cdots} f(x|u_1, \ldots, u_n) \cdot \prod_{i=1}^{n} f(u_i|e^+_X) \, du_1 \cdots du_n
\]
a weighted sum of Gaussians. We refer the reader to (Driver 1995) for a complete derivation. We write \( \lambda(x) \) in the following form:

\[
\lambda(x) = \sum_{j=1}^{M_n} \alpha_{j_0} N \left( x; \sigma_{j_0}^2, \mu_{j_0} \right)
\]

(12)

where \( M_0 = \prod_{i=1}^{M} a_i \), \{\alpha_{j_0}\} is a set of real valued weights, \{\{\sigma_{j_0}^2\}\} is a set of real valued variances, and \{\mu_{j_0}\} is a set of real valued means.

We can now compute \( BEL(x) \) by forming the product of (12) and (9). \( BEL(x) \) can then be written as a weighted sum of Gaussians by using (5):

\[
BEL(x) = \alpha \sum_{j=1}^{M} \sum_{j_0=1}^{M_0} \eta_{j,j_0} N \left( x; \frac{\sigma_{j_0}^2}{\sigma_j^2 + \sigma_{j_0}^2}, \frac{\mu_{j_0} - \mu_j}{\sigma_j^2 + \sigma_{j_0}^2} \right)
\]

(13)

where

\[
\eta_{j,j_0} = \gamma_j \alpha_{j_0} N \left( \mu_j; \sigma_j^2 + \sigma_{j_0}^2, \mu_{j_0} \right)
\]

and \( \alpha \) is a normalization constant chosen so that \( \int BEL(x) \, dx = 1 \). Integrating (13) with respect to \( x \), it follows that

\[
\alpha = \left( \sum_{j=1}^{M} \sum_{j_0=1}^{M_0} \eta_{j,j_0} \right)^{-1}
\]

(15)

### 3.2 TOP DOWN PROPAGATION

The message \( \pi_{Y_j}(x) \) that node \( X \) sends to its \( j \)th child \((j = 1, 2, \ldots, m)\) is formed as follows:

\[
\pi_{Y_j}(x) \defeq f(x; e_j - e_{j_0}) = BEL(x; e_j \neq \emptyset)
\]

(16)

So \( \pi_{Y_j}(x) \) can be computed by the method of the last section with the assumption that \( \lambda_{Y_j}(x) = 1 \).

### 3.3 BOTTOM UP PROPAGATION

The message \( \lambda_X(u_i) \) that node \( X \) sends to its \( i \)th parent \((i = 1, 2, \ldots, n)\) is formed as follows:

\[
\lambda_X(u_i) \defeq f(e_i; e_i^+|u_i)
\]

(17)
Consider the first distribution in the integrand:

\[ f(e - e^+|u_1, \ldots, u_n, x) = f(e_1^+ \ldots, e_m^+, e_{i-1}^+, e_{i+1}^+, \ldots, e_n^+| u_1, \ldots, u_n, x) \]

\[ = f(e_1^+, \ldots, e_m^+|x) \cdot f(e_1^+, \ldots, e_{i-1}^+, e_{i+1}^+, \ldots, e_n^+| u_1, \ldots, u_n, x) \]

\[ = \lambda(x) \prod_{k=1 \atop k \neq i}^{n} f(e_k^+|u_k) \cdot \frac{f(e_i^+)}{f(u_k)} \]

\[ = C(\lambda(x)) \prod_{k=1 \atop k \neq i}^{n} \pi_X(u_k) \prod_{k=1 \atop k \neq i}^{n} \frac{1}{f(u_k)} \]  

(18)

Where \( C = \prod_{k=1 \atop k \neq i}^{n} f(e_k^+) \) is constant for a given set of evidence. Next, consider the second distribution in the integrand:

\[ f(u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_n, x|u_i) = f(x|u_1, \ldots, u_n)f(u_i, \ldots, u_{i-1}, u_{i+1}, \ldots, u_n|u_i) \]

\[ = f(x|u_1, \ldots, u_n) \prod_{k=1 \atop k \neq i}^{n} f(u_k) \]  

(19)

Substituting (18) and (19) into (17), we have:

\[ \lambda_X(u_i) = C \int_{u_1} \int_{u_{i-1}} \int_{u_{i+1}} \ldots \int_{u_n} \int_{x} \lambda(x) \cdot \prod_{k=1 \atop k \neq i}^{n} \pi_X(u_k)f(x|u_1, \ldots, u_n) \cdot dx \cdot du_1 \cdot \ldots \cdot du_{i-1} \cdot du_{i+1} \cdot \ldots \cdot du_n \]  

(20)

The constant \( C \) will get absorbed during the normalization of \( BEL(x) \), so we can ignore it. \( f(x|u_1, \ldots, u_n) \), \( \lambda(x) \), and \( \pi_X(u_k) \) are given by (4), (12), and (6) respectively. Substituting these into (20) we have:

\[ \lambda_X(u_i) = \int_{u_1} \int_{u_{i-1}} \int_{u_{i+1}} \ldots \int_{u_n} \int_{x} \sum_{j=0}^{M_o} \sum_{j_1=1}^{M_1} \sum_{j_{i-1}=1}^{M_{i-1}} \sum_{j_{i+1}=1}^{M_{i+1}} \ldots \sum_{j_{n}=1}^{M_n} \alpha_{j_0} \cdot N(x; (\sigma_{\lambda,j_0})^2, \mu_{j_0}) \cdot \prod_{k=1 \atop k \neq i}^{n} \sum_{j_k=1}^{M_k} \alpha_{j_k} \cdot N(u_k; (\sigma_{\tau,j_k})^2, \mu_{\tau,j_k}) \cdot \prod_{j=1}^{m} \sum_{j=1}^{M_j} \alpha_{j} \cdot N(u_j; (\sigma_{\sigma,j}^2)^2, \mu_{\sigma,j}^2) \cdot \prod_{k=1 \atop k \neq i}^{n} \sum_{j_k=1}^{M_k} \alpha_{j_k} \cdot N(u_k; (\sigma_{\tau,j_k})^2, \mu_{\tau,j_k}) \cdot dx \cdot du_1 \cdot \ldots \cdot du_{i-1} \cdot du_{i+1} \cdot \ldots \cdot du_n \]  

(21)

The constant \( C \) will get absorbed during the normalization of \( BEL(x) \), so we can ignore it. \( f(x|u_1, \ldots, u_n) \), \( \lambda(x) \), and \( \pi_X(u_k) \) are given by (4), (12), and (6) respectively. Substituting these into (20) we have:

\[ \lambda_X(u_i) = \int_{u_1} \int_{u_{i-1}} \int_{u_{i+1}} \ldots \int_{u_n} \int_{x} \sum_{j=0}^{M_o} \sum_{j_1=1}^{M_1} \sum_{j_{i-1}=1}^{M_{i-1}} \sum_{j_{i+1}=1}^{M_{i+1}} \ldots \sum_{j_{n}=1}^{M_n} \alpha_{j_0} \cdot N(x; (\sigma_{\lambda,j_0})^2, \mu_{j_0}) \cdot \prod_{k=1 \atop k \neq i}^{n} \sum_{j_k=1}^{M_k} \alpha_{j_k} \cdot N(u_k; (\sigma_{\tau,j_k})^2, \mu_{\tau,j_k}) \cdot \prod_{j=1}^{m} \sum_{j=1}^{M_j} \alpha_{j} \cdot N(u_j; (\sigma_{\sigma,j}^2)^2, \mu_{\sigma,j}^2) \cdot \prod_{k=1 \atop k \neq i}^{n} \sum_{j_k=1}^{M_k} \alpha_{j_k} \cdot N(u_k; (\sigma_{\tau,j_k})^2, \mu_{\tau,j_k}) \cdot dx \cdot du_1 \cdot \ldots \cdot du_{i-1} \cdot du_{i+1} \cdot \ldots \cdot du_n \]  

(22)

Note that if \( \lambda(x) \) is constant then from (20) we have:

\[ \lambda_X(u_i) = C \int_{u_1} \int_{u_{i-1}} \int_{u_{i+1}} \ldots \int_{u_n} \int_{x} \sum_{j=0}^{M_o} \sum_{j_1=1}^{M_1} \sum_{j_{i-1}=1}^{M_{i-1}} \sum_{j_{i+1}=1}^{M_{i+1}} \ldots \sum_{j_{n}=1}^{M_n} \alpha_{j_0} \cdot N(x; (\sigma_{\lambda,j_0})^2, \mu_{j_0}) \cdot \prod_{k=1 \atop k \neq i}^{n} \sum_{j_k=1}^{M_k} \alpha_{j_k} \cdot N(u_k; (\sigma_{\tau,j_k})^2, \mu_{\tau,j_k}) \cdot \prod_{j=1}^{m} \sum_{j=1}^{M_j} \alpha_{j} \cdot N(u_j; (\sigma_{\sigma,j}^2)^2, \mu_{\sigma,j}^2) \cdot \prod_{k=1 \atop k \neq i}^{n} \sum_{j_k=1}^{M_k} \alpha_{j_k} \cdot N(u_k; (\sigma_{\tau,j_k})^2, \mu_{\tau,j_k}) \cdot dx \cdot du_1 \cdot \ldots \cdot du_{i-1} \cdot du_{i+1} \cdot \ldots \cdot du_n \]  

(23)
Driver and Morrell

\[ \cdot dxdu_1 \cdots du_{i-1}du_{i+1} \cdots du_n \]
\[ = C \prod_{k=1}^{n} \int_{u_k}^{u_k} \pi_X(u_k)du_k \]
\[ = \text{constant} \]

Hence, if \( \lambda(x) = 1 \) then \( \lambda_X(u_i) = 1 \) for each \( i \). So just like with discrete variables, evidence gathered at a node does not affect any spousal nodes until their common child node obtains evidence.

### 3.4 BOUNDARY CONDITIONS

If \( X \) is a root node (i.e. a node with no parents) that has not been instantiated, then we set \( \pi(x) \) equal to the prior density function \( f(x) \). This prior distribution is approximated by a sum of weighted Gaussians.

If \( X \) is a leaf node (i.e. a node with no children) that has not been instantiated, then we set \( \lambda_X(u_i) = 1 \). This implies that \( \text{BEL}(x) = \pi(x) \).

If \( X \) is an evidence node, say \( X = x_0 \), then we set \( \lambda_X(u_i) = \delta(x - x_0) = N(x; 0, x_0) \) regardless of the incoming \( \lambda \)-messages. This implies that \( \text{BEL}(x) = N(x; 0, x_0) \) as we would expect. Furthermore, for each \( j \), \( \pi_Y(x) = N(x; 0, x_0) \) is the message that node \( X \) sends to its children (each child gets the same message in this case). The messages that node \( X \) sends to its parents are still weighted sums of Gaussians.

### 3.5 SPECIAL CASE: LINEARITY

In this section, we consider the special case when the function \( g \) of Equation (1) has the form

\[ g(U_1, U_2, \ldots, U_n) = b_1 U_1 + b_2 U_2 + \cdots + b_n U_n \quad (23) \]

where the \( b_i \)'s are real numbers. This relationship was suggested in (Pearl 1988). With this we have

\[ f(x|u_1, \ldots, u_n) = N \left( x; \sigma^2_{w_X}, \sum_{i=1}^{n} b_i u_i \right). \quad (24) \]

We note that only the computations of \( \lambda(x) \) and \( \lambda_X(u_i) \) involve the distribution \( f(x|u_1, \ldots, u_n) \), so all other computations are the same as before. Furthermore, we can obtain closed form solutions for \( \lambda(x) \) and \( \lambda_X(u_i) \) without invoking the approximation given by (4). To do so, we use the following identity:

\[ \int \ldots \int \pi_n \left( \prod_{i=1}^{n} N(x_i; \sigma_i^2, \mu_i) \right) \cdot N \left( \sum_{i=1}^{n} b_i x_i; \sigma^2, \mu \right) \, dx_1 \cdots dx_n \]
\[ = N \left( \mu; \sigma^2 + \sum_{i=1}^{n} b_i^2 \sigma_i^2, \sum_{i=1}^{n} b_i \mu_i \right) \quad (25) \]

For completeness we state the final result but omit the proof; we refer the interested reader to (Driver 1995):

\[ \pi(x) = \sum_{k_1=1}^{M_1} \cdots \sum_{k_n=1}^{M_n} \left( \prod_{i=1}^{n} \alpha_{i|k_i} \right) \cdot N \left( x; \sigma^2_{w_X} + \sum_{i=1}^{n} b_i^2 (\sigma_{x,i}^2)^2, \sum_{i=1}^{n} b_i \mu_{x,i} \right) \quad (26) \]

\[ \lambda_X(u_i) = \sum_{j_0=1}^{M_0} \sum_{j_1=1}^{M_1} \cdots \sum_{j_{n-1}=1}^{M_{n-1}} \sum_{j_n=1}^{M_n} \alpha_{j_0} \sum_{k \neq i}^{M_k} \left( \prod_{k=1}^{n} \alpha_{j_k}^k \right) N \left( u_i; \frac{1}{b_i}, \sigma^2_{w_X} + (\sigma_{x,j_0}^2 + \sum_{k \neq i}^{M_k} b_k^2 (\sigma_{x,j_k}^2)^2) \right) \quad (27) \]

Note that the means and variances of the Normal distributions in (26) and (27) have the same form as those in Pearl’s result. The only real difference here is that our result is a sum of weighted Gaussians and Pearl’s result is a single Gaussian.

### 4 EXAMPLE

In this section, we present a simple example that illustrates the characteristics of this approach to continuous Bayesian networks. The example uses the network of Figure 2. Nodes \( X \) and \( Y \) are independent and uniformly distributed on \([0, 1]\). Node \( Z \) is related to \( X \) and \( Y \) by the following equation:

\[ Z = X + Y + w_Z, \]

where \( w_Z \) is a Gaussian random variable with zero mean and variance 0.01; \( w_Z \) is independent of \( X \) and \( Y \).

Figure 3 shows a graph of our approximation to the prior distributions of nodes \( X \) and \( Y \); this approximation consists of 20 Gaussians with uniformly spaced.
means, equal variances, and equal weights. The absolute error in this approximation is 9%. Other approximations could be found that provide a better representation of the densities with fewer Gaussians; potential methods by which this can be done are discussed in Section 5.

Performing the updating algorithm with no evidence in the network, we obtain the distribution for node $Z$. This is shown in Figure 4. From probability theory, we know that the distribution of a sum of two independent random variables is the convolution of the individual distributions. So theoretically, the distribution of $Z$ should be the triangular function on the interval $[0, 2]$. Taking into account the effects of the noise $w_Z$ and the approximation error in the prior distributions, this is exactly what we have in Figure 4.

Next, we assume that node $X$ is instantiated at a value of 1.0. Performing the updating algorithm for this evidence, we obtain the belief function for node $Z$. This is shown in Figure 5. We can explain this as follows. Given $X = 1.0$, $Z$ is just the sum of the constant 1.0 and a uniform distribution (plus the noise term $w_Z$). Hence, $Z$ is a uniform distribution on $[1, 2]$. From Figure 6, we see that our results are a reasonable approximation to a delta function, especially when we consider the effects of the noise and the approximation error in the prior density functions.

5 GAUSSIAN SUM APPROXIMATIONS

In this section, we discuss briefly some of the issues involved in choosing sum-of-Gaussian approximations to probability density functions. We first discuss the existence of good approximations. We then mention several techniques that can be used to obtain these approximations.

Let $S$ be a compact subset of $\mathcal{R}^n$ and consider the set:

$$G_S = \left\{ g \in C[S] \left| g(x) = \sum_{i=1}^{m} c_i \exp \left[ \frac{-1}{2\sigma_i^2} (x - \mu_i)^T (x - \mu_i) \right] \right. ; m \in \mathcal{N}; c_i, \sigma_i \in \mathcal{R}; \mu_i \in \mathcal{R}^n \right\}$$

(28)
where $C[S]$ is the set of all continuous functions from $S$ to $R$. By a simple application of the Stone-Weierstrass theorem, it can be shown (Girosi 1990) that $G_s$ is dense in $C[S]$. That is, given any $f \in C[S]$ and any $\varepsilon > 0$, there exists $g \in G_s$ such that $\int |g(x) - f(x)| d\nu < \varepsilon$. (Note: here we have used the $L_1$ norm, in fact the result holds for any $L_p$ norm, $1 \leq p < \infty$.) In other words, any arbitrary function in $C[S]$ can be approximated arbitrarily well by a finite sum of weighted Gaussians.

In this paper, the functions we are approximating are probability density functions. Since density functions die off at infinity, as an approximation we may assume that they are defined on compact subsets of $R^n$ (i.e. we set them to zero outside some compact subset of $R^n$). Hence, by the above results it is reasonable to assume that any density function can be approximated arbitrarily well by a finite sum of Gaussians.

We now turn our attention to actually finding such approximations. One approach to obtaining approximations is the use of neural networks (Poggio 1990). Another method (Klopfenstein 1983) approximates an arbitrary function by uniformly spaced Gaussians; using Fourier transform techniques, error estimates can be obtained. Other methods include simulated annealing and gradient descent algorithms. Our best results have been with the gradient descent algorithm, especially when approximating functions with a small number of variables.

Acknowledgements

This work supported by the United States Army Research Office, grant number DAAH04-93-G-0218

References

J. Pearl, *Probabilistic Reasoning In Intelligent Systems*. San Mateo, CA: Morgan Kaufmann Publishers, 1988.

R. D. Shachter and C. R. Kenley, "Gaussian Influence Diagrams," *Management Science*, Vol. 35, No. 5, May 1989, pp. 527–550.

S. L. Lauritzen and N. Wermuth, "Graphical Models for Associations Between Variables, Some of Which are Qualitative and Some Quantitative," *The Annals of Statistics*, Vol. 17, No. 1, 1989, pp. 31–57.

S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H. G. Leimer, "Independence Properties of Directed Markov Fields," *Networks*, Vol. 20, Iss. 5, Aug 1990, pp. 491–505.

D. J. Spiegelhalter and S. L. Lauritzen, "Sequential Updating of Conditional Probabilities on Directed Graphical Structures," *Networks*, Vol. 20, Iss. 5, Aug 1990, pp. 579–605.

E. Driver and D. Morrell, "Implementation of Continuous Bayesian Networks Using Sums of Weighted Gaussians," Technical Report TRC-SP-DRM-9501, Electrical Engineering Department, ASU, May 1995.

F. Girosi and T. Poggio, "Networks and the Best Approximation Property," *Biological Cybernetics*, Vol. 63, 1990, pp. 169–176.

T. Poggio and F. Girosi, "Networks for Approximation and Learning," *Proceedings of the IEEE*, Vol. 78, No. 9, Sep. 1990, pp. 1481–1497.

R. Klopfenstein and R. Sverdlove, "Approximation by Uniformly Spaced Gaussian Functions," in *Approximation Theory IV*, (C. K. Chui, L. L. Schumaker, and J. D. Ward, eds.), pp. 575–580, Academic Press, 1983.