Appendices

2.6 Chapter 2 Appendix

2.6.1 Multivariate Gaussian Sampling

Let $\mathbf{X} = \{X_1, X_2, ..., X_N\}$ be a random vector that follows multivariate Gaussian distribution, i.e., $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Directly sampling from the multivariate Gaussian distribution may be challenging. This challenge can be mitigated with the re-parameterization trick. Let $\mathbf{Z}$ be a random vector that follows the standard multivariate Gaussian distribution, i.e., $\mathbf{Z} \sim \mathcal{N}(0, \mathbf{I})$, where $\mathbf{I}$ is the identity covariance matrix. We can easily prove that $\mathbf{x} = \boldsymbol{L}\mathbf{z} + \boldsymbol{\mu}$, where $\boldsymbol{L}$ is the lower triangle matrix resulted from Cholesky decomposition of $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$. As $\mathbf{Z}$ follows standard multivariate Gaussian distribution, we can sample each element of $\mathbf{Z}$ independently, yielding a sample $\mathbf{z}_s$, based on which we obtain the corresponding sample for $\mathbf{X}$ as $\mathbf{x}_s = \boldsymbol{L}\mathbf{z}_s + \boldsymbol{\mu}$.

3.9 Chapter 3 Appendix

3.9.4 Inference under uncertain evidence

For a Bayesian Network (BN), we can perform probability inference for a query node X_Q given an evidence $X_E = x_e$ by computing $P(X_Q|X_E = x_E)$. However, the conventional inference method assumes there exists no uncertainty in the evidence. For many real world applications, the evidence we observe contains uncertainty, which may originate from noise in the data or from the imprecision with the evidence measuring device. We refer such evidence as *uncertain evidence*. Based on how the the uncertainty of the evidence is interpreted and represented, we can classify the uncertain evidence into two categories: soft evidence and virtual evidence.

Let X_E be the uncertain evidence variable. Soft evidence captures the uncertainty of X_E by a probability $q(X_E)$, while the virtual evidence encodes the uncertainty in X by it's likelihood ratio with respect to a virtual binary variable Z. Inference with soft evidence can be performed using Jeffery's rule [1] as in Eq. 3.56, that is,

$$p(\mathbf{x}_Q|q(\mathbf{x}_E)) = \sum_{\mathbf{x}_E} p(\mathbf{x}_Q|\mathbf{x}_E)q(\mathbf{x}_E) \tag{3.185}$$

For virtual evidence, according to Judea Pearl [2], we can introduce a virtual node Z as a child of X_E. Z has the same states as node X_E. The CPT for node Z is partially specified by the likelihood ratio of X_E, i.e., $\frac{p(Z=z|X_E=x_E)}{p(Z=z|X_E \neq x_E)}$. Given this specification, we can then perform inference of $p(X_Q|Z = z)$. Further details about these two types of uncertain evidences can be found in [3, 4, 5, 6, 7, 8].

3.9.5 Hamiltonian Monte-Carlo (HMC) Sampling

The benefit of the MH algorithm is that it allows sampling from an un-normalized probability. Despite it's strengths, traditional MH algorithm uses a simple proposal distribution, largely based on random walk, such as a Gaussian distribution; it is hence inefficient and cannot scale up well to high dimensional space. HMC [9] was introduced to address this limitation by employing Hamiltonian dynamics to speed up proposal generation. In a Hamilton dynamical system, denote $\boldsymbol{x}$ as the position, r as the momentum, t as the time, $H = H(\boldsymbol{x}, r, t)$ is the Hamiltonian function that captures the total energy of the system and it satisfies the Hamiltonian equations:

$$\frac{dr}{dt} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}, \quad \frac{\mathrm{d}\boldsymbol{x}}{dt} = +\frac{\partial \mathcal{H}}{\partial r} \tag{3.186}$$

For a closed system, the total energy equals to sum of the potential energy and the kinetic energy. Denote the random variables that we want to sample as $\mathbf{x}$; the position $\boldsymbol{x}$ in Eq. (3.186) is now replaced by $\mathbf{x}$. Define the potential energy function as $U(\mathbf{x}) = -\log p(\mathbf{x})$ and the kinetic energy function as $K(r) = \frac{1}{2} r^T \Sigma^{-1} r$, the Hamiltonian function can be written as

$$H(\mathbf{x}, r) = U(\mathbf{x}) + K(r) = -\log p(\mathbf{x}) + \frac{1}{2} r^T \Sigma^{-1} r \tag{3.187}$$

where we assume r follows a Gaussian distribution $\mathcal{N}(0, \Sigma)$. Combining Eq. (3.186) and Eq. (3.187), we can derive the update equations for the proposal in HMC. In a discretized version, denote the step size as ϵ, the iteration number as i, and we have the update equations:

$$\begin{aligned} r_i &= r_{i-1} - \epsilon \nabla U\left(\mathbf{x}_{i-1}\right) \\ \mathbf{x}_i &= \mathbf{x}_{i-1} + \epsilon \nabla K\left(r_i\right) = \mathbf{x}_{i-1} + \epsilon \Sigma^{-1} r_i \end{aligned} \tag{3.188}$$

The update equation guarantees next sample has a higher probability than the current sample. Algorithm 3.15 provides the pesudo-code for the HMC sampling method. Further information about the HMC method may be found in [10].

Algorithm 3.15 Hamiltonian Monte Carlo algorithm

```
Input: Unnormalized target distribution $\tilde{p}(\mathbf{x})$, momentum distribution $p(r) = \mathcal{N}(0, \Sigma)$, potential energy function $U(\mathbf{x}) = -\log \tilde{p}(\mathbf{x})$, the total energy function $U(\mathbf{x}) + \frac{1}{2} r^T \Sigma^{-1} r$, and step size $\epsilon$.
Initialize: Starting position $\mathbf{x}_0$ at $t = 0$.
for $t = 0, 1, 2, \dots$ do
    $r^t \sim \mathcal{N}(0, \Sigma)$;                                   ▷ Sample momentum $r^t$
    $r^t \leftarrow r^t - \frac{\epsilon}{2} \nabla U(\mathbf{x}^t)$;
    for $i = 1, 2, \dots m$ do                        ▷ Simulate discretized Hamiltonian dynamics
        $\mathbf{x}_i \leftarrow \mathbf{x}_{i-1} + \epsilon \Sigma^{-1} r_{i-1}$;
        $r_i \leftarrow r_{i-1} - \epsilon \nabla U(\mathbf{x}_i)$;
    end for
    $r_m \leftarrow r_m - \frac{\epsilon}{2} \nabla U(\mathbf{x}_m)$;
    set $(\mathbf{x}', r') = (\mathbf{x}_m, r_m)$;
    $A(\mathbf{x}', \mathbf{x}^t) = \min\left(1, e^{H(\mathbf{x}^t, r^t) - H(\mathbf{x}', r')}\right)$;   ▷ Compute the acceptance probability
    $u \sim \mathcal{U}(0, 1)$;                           ▷ Generate a uniform random number
    if $u \leq A(\mathbf{x}', \mathbf{x}^t)$ then $\mathbf{x}^{t+1} = \mathbf{x}'$;            ▷ Accept $\mathbf{x}'$ based on $A(\mathbf{x}', \mathbf{x}^t)$
    else $\mathbf{x}^{t+1} = \mathbf{x}^t$;                      ▷ Reject $\mathbf{x}'$, and use the old state
    end if
end for
```

The benefit of HMC algorithm compared to traditional MH algorithm is that the dynamics speeds up inference because the momentum r of the system prevents the random walk behavior. Distances between successively generated proposal points from HMC are typically large, so we need fewer iterations to obtain representative samples. And since the proposal distribution moves towards the direction that maximizes the target distribution, HMC in most cases accepts new states. To summarize, by explicitly exploiting the Hamiltonian dynamics, HMC is significantly more efficient than traditional MH algorithms.

3.9.6 BN Belief propagation examples

Given the structure and parameters of a Bayesian Network as shown in Figure 3.48, we perform detailed step-by-step calculations to perform the the sum-product and max-product inference.

3.9.6.1 Sum-product BP examples

We first show the process for sum-product inference given a boundary evidence. This is then followed by sum-product inference given an non-boundary evidence.

Sum-product inference given boundary evidence

We perform the sum-product inference given the evidence: `M`(Marry Calls) = '1'. Particularly, we follow the order of nodes below for message passing. Please note this is not necessary the optimal order.

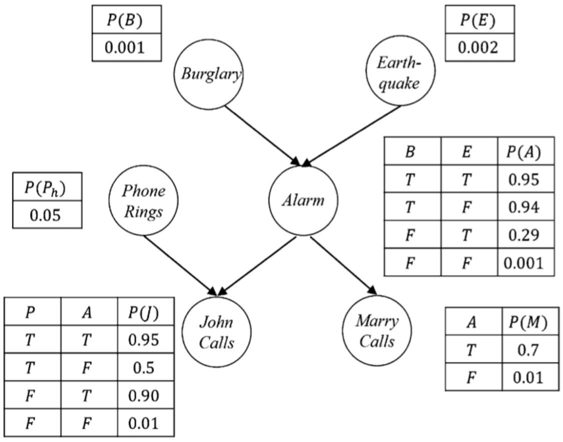

Figure 3.48: An example Bayesian Network

1. Node A (Alarm) collects all messages from its children M and J and its parents E and B, updates its belief, and normalizes
2. Node B (Burglary) collects its message from child A, updates its belief, and normalizes
3. Node E (Earthquake) collects its message from child A, updates its belief, and normalizes
4. Node J (John Calls) collects its messages from it's parents Ph and A, updates its belief, and normalizes
5. Node Ph (Phone Rings) collects its message from child J, updates its belief, and normalizes
6. Node M (Marry Calls) collects its message from parent A, updates its belief, and normalizes (optional)

We first initialize the messages for the boundary nodes and the evidence nodes as shown in Figure 3.49. The messages for the remaining nodes are initialized to ones. In addition, the incoming messages to each node (including the evidence node) are initialized to ones. For the evidence node M, we need revise the entries of the CPTs involving node M as follows: the CPT entries corresponding to $m \neq 1$ are set to zeros, i.e., $p(m = 0|a) = 0$ and the entries corresponding to $m = 1$, i.e., $p(m = 1|a)$, remain unchanged. Below, we show the detailed calculation of messages and belief updating of each node using Eqs. (3.23-3.26) for the first iteration.

Message computation and belief updating for node A

The message node A receives from its parent E is

$$\pi_e(a) = \pi(e) \prod_{c \in \text{Child}(e) \backslash a} \lambda_c(e) = \pi(e) = [0.002; 0.998] \tag{3.189}$$

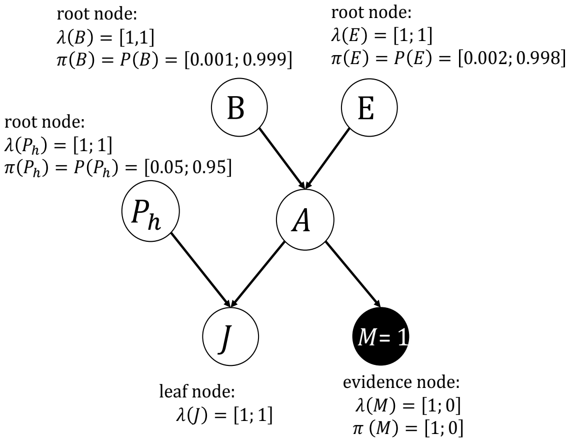

Figure 3.49: Initialization of messages for sum of product inference with a boundary evidence.

The message node A receives from its parent B is

$$\pi_b(a) = \pi(b) \prod_{c \in \text{Child}(b) \backslash a} \lambda_c(b) = \pi(b) = [0.001; 0.999] \tag{3.190}$$

The total message node A receives from its parents nodes E and B is

$$\begin{aligned} \pi(a) &= \sum_{b,e} p(a|b,e)\pi_e(a)\pi_b(a) \\ &= \begin{bmatrix} \sum_{b,e} p(a=1|b,e)\pi_e(a)\pi_b(a) \\ \sum_{b,e} p(a=0|b,e)\pi_e(a)\pi_b(a) \end{bmatrix} = \begin{bmatrix} 0.002516 \\ 0.997484 \end{bmatrix} \end{aligned} \tag{3.191}$$

The message node A receives from its child J is

$$\begin{aligned} \lambda_j(a) &= \sum_j \lambda(j) \sum_{p_h} p(j|a,p_h)\pi_{p_h}(j) \\ &= \begin{bmatrix} \sum_j \lambda(j) \sum_{p_h} p(j|a=1,p_h)\pi_{p_h}(j) \\ \sum_j \lambda(j) \sum_{p_h} p(j|a=0,p_h)\pi_{p_h}(j) \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \end{aligned} \tag{3.192}$$

where $\pi_{p_h}(j) = [1;1]$ as initialized.

The message node A receives from its child M is

$$\begin{aligned} \lambda_m(a) &= \sum_m \lambda(m)p(m|a) \\ &= \begin{bmatrix} \sum_m \lambda(m)p(m|a=1) \\ \sum_m \lambda(m)p(m|a=0) \end{bmatrix} = \begin{bmatrix} 0.7 \\ 0.01 \end{bmatrix} \end{aligned} \tag{3.193}$$

The total message node A receives from its children nodes J and M is

$$\lambda(a) = \lambda_j(a)\lambda_m(a) = \begin{bmatrix} 1.4 \\ 0.02 \end{bmatrix} \tag{3.194}$$

Given current messages, we obtain the normalized belief of node A:

$$Bel(a) = \alpha\pi(a)\lambda(a) = \begin{bmatrix} 0.150068 \\ 0.849932 \end{bmatrix} \tag{3.195}$$

Message computation and belief updating for node B

The message node B receives from its child A is

$$\begin{aligned} \lambda_a(b) &= \sum_a \lambda(a) \sum_e p(a|b,e)\pi_e(a) \\ &= \begin{bmatrix} \sum_a \lambda(a) \sum_e p(a|b=1,e)\pi_e(a) \\ \sum_a \lambda(a) \sum_e p(a|b=0,e)\pi_e(a) \end{bmatrix} = \begin{bmatrix} 1.317228 \\ 0.022178 \end{bmatrix} \end{aligned} \tag{3.196}$$

where $\pi_e(a) = [0.002; 0.998]$ is calculated in Eq. 3.189. The total message node B receives from its child A is

$$\lambda(b) = \lambda_a(b) = \begin{bmatrix} 1.317228 \\ 0.022178 \end{bmatrix} \tag{3.197}$$

Given current messages, we obtain the normalized belief of node B:

$$Bel(b) = \alpha\pi(b)\lambda(b) = \begin{bmatrix} 0.056117 \\ 0.943883 \end{bmatrix} \tag{3.198}$$

Message computation and belief updating for node E

The message node E receives from its child A is

$$\begin{aligned} \lambda_a(e) &= \sum_a \lambda(a) \sum_b p(a|b,e)\pi_b(a) \\ &= \begin{bmatrix} \sum_a \lambda(a) \sum_b p(a|b,e=1)\pi_b(a) \\ \sum_a \lambda(a) \sum_b p(a|b,e=0)\pi_b(a) \end{bmatrix} = \begin{bmatrix} 0.421111 \\ 0.022676 \end{bmatrix} \end{aligned} \tag{3.199}$$

where $\pi_b(a) = [0.001; 0.999]$ is calculated in Eq. 3.190.

The total message node E receives from its child A is

$$\lambda(e) = \lambda_a(e) = \begin{bmatrix} 0.421111 \\ 0.022676 \end{bmatrix} \tag{3.200}$$

Given current messages, we obtain the normalized belief of node E

$$Bel(e) = \alpha\pi(e)\lambda(e) = \begin{bmatrix} 0.035881 \\ 0.964119 \end{bmatrix} \tag{3.201}$$

Message computation and belief updating for node J

The message node J receives from its parent A is

$$\pi_a(j) = \pi(a) \prod_{c \in \text{Child}(a) \backslash j} \lambda_c(a) = \pi(a)\lambda_m(a) = \begin{bmatrix} 0.001761 \\ 0.009975 \end{bmatrix} \tag{3.202}$$

where $\lambda_m(a) = [0.7; 0.01]$ is calculated in Eq. 3.193.

The message node J receives from its parent Ph is

$$\pi_{p_h}(j) = \pi(p_h) \prod_{c \in \text{Child}(p_h) \backslash j} \lambda_c(p_h) = \pi(p_h) = \begin{bmatrix} 0.05 \\ 0.95 \end{bmatrix} \tag{3.203}$$

The total message node J receives from its parent nodes Ph and A is

$$\begin{aligned} \pi(j) &= \sum_{p_h, a} p(j|p_h, a)\pi_{p_h}(j)\pi_a(j) \\ &= \begin{bmatrix} \sum_{p_h,a} p(j = 1|p_h, a)\pi_{p_h}(j)\pi_a(j) \\ \sum_{p_h,a} p(j = 0|p_h, a)\pi_{p_h}(j)\pi_a(j) \end{bmatrix} = \begin{bmatrix} 0.001933 \\ 0.009803 \end{bmatrix} \end{aligned} \tag{3.204}$$

where $\pi_a(j) = [0.001761; 0.009975]$ is calculated in Eq. 3.202 and $\pi_{p_h}(j) = [0.05; 0.95]$ is calculated in Eq. 3.203.

Given current messages, we obtain the normalized belief of node J

$$Bel(j) = \alpha\pi(j)\lambda(j) = \begin{bmatrix} 0.164707 \\ 0.835293 \end{bmatrix} \tag{3.205}$$

Message computation and belief updating for node Ph

The message node Ph receives from its child J is

$$\begin{aligned} \lambda_j(p_h) &= \sum_j \lambda(j) \sum_a p(j|p_h, a)\pi_a(j) \\ &= \begin{bmatrix} \sum_j \lambda(j) \sum_a p(j|p_h = 1, a)\pi_a(j) \\ \sum_j \lambda(j) \sum_a p(j|p_h = 0, a)\pi_a(j) \end{bmatrix} = \begin{bmatrix} 0.011736 \\ 0.011736 \end{bmatrix} \end{aligned} \tag{3.206}$$

where $\pi_a(j) = [0.001761; 0.009975]$ is calculated in Eq. 3.202.

The total message node Ph receives from its child J is

$$\lambda(p_h) = \lambda_j(p_h) = \begin{bmatrix} 0.011736 \\ 0.011736 \end{bmatrix} \tag{3.207}$$

Given current messages, we obtain the normalized belief of node Ph

$$Bel(p_h) = \alpha\pi(p_h)\lambda(p_h) = \begin{bmatrix} 0.0500 \\ 0.9500 \end{bmatrix} \tag{3.208}$$

Message computation and belief updating for node M

$$\pi_a(m) = \pi(a) \prod_{c \in \text{Child}(a) \setminus m} \lambda_c(a) = \pi(a)\lambda_j(a) = \begin{bmatrix} 0.005032 \\ 1.994968 \end{bmatrix} \tag{3.209}$$

where $\lambda_j(a) = [2; 2]$ is calculated in Eq. 3.192. The total message node M receives from its parent A is

$$\pi(m) = \sum_a p(m|a)\pi_a(m) = \begin{bmatrix} 0.023472 \\ 0 \end{bmatrix} \tag{3.210}$$

where $p(m = 0|a) = 0$ as we revise given the evidence $m = 1$. Given current messages, we obtain the normalized belief of node M

$$Bel(m) = \alpha\pi(m)\lambda(m) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.211}$$

where $\lambda(m) = [1; 0]$ as initialized. Since node M is the evidence node, its belief doesn't change over iterations. Note updating belief for node M is optional as it is a boundary evidence node.

We now finish the first iteration. We repeat the above process for the second and third iteration. During each iteration, for each node, we update its messages based on the current messages the node receives from its parents and children. Comparing the messages and beliefs from the second iteration and the third iteration, we can observe that there is no change, i.e., the belief propagation converges after the second iteration. In the end, we obtain the belief of each node given the evidence as

$$\begin{aligned} Bel(a) &= p(a|m = 1) = [0.150068; 0.849932] \\ Bel(b) &= p(b|m = 1) = [0.056117; 0.943883] \\ Bel(e) &= p(e|m = 1) = [0.035881; 0.964119] \\ Bel(j) &= p(j|m = 1) = [0.164707; 0.835293] \\ Bel(p_h) &= p(p_h|m = 1) = [0.050000; 0.950000] \end{aligned} \tag{3.212}$$

Sum-product inference given non-boundary evidence We now consider another example where we are given a non-boundary evidence `Alarm = '1'`. The initialization of the boundary nodes, the non-boundary nodes, and evidence nodes remain the same as we did for the boundary evidence case as shown in Figure 3.50. In addition, the incoming messages to each node are initialized to ones. For the evidence A, the entries of CPTs involving node A are revised as follows: the entries of the CPTs corresponding to $a \neq 1$ are set to zeros, i.e., $p(a = 0|b, e) = 0$, $p(j|p_h, a = 0) = 0$ and $p(m|a = 0) = 0$, while the entries of CPTs corresponding to $a = 1$, i.e., $p(a = 1|b, e)$, $p(j|p_h, a = 1)$ and $p(m|a = 1)$, remain unchanged as the original conditional probabilities. Given the initialization, the belief propagation process remains the same. In the following, we perform the sum-product inference given an non-boundary evidence `Alarm = 1` for the same example shown in Figure 3.48 and following the same node order.

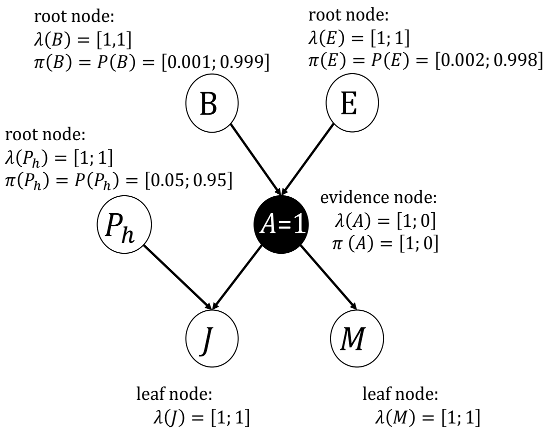

Figure 3.50: Initialization of messages for sum of product inference with a non-boundary evidence node.

Message computation and belief updating for node A [1]

The message node A receives from its parent E is

$$\pi_e(a) = \pi(e) \prod_{c \in \text{Child}(e) \backslash a} \lambda_c(e) = \pi(e) = [0.002; 0.998] \tag{3.213}$$

The message node A receives from its parent B is

$$\pi_b(a) = \pi(b) \prod_{c \in \text{Child}(b) \backslash a} \lambda_c(b) = \pi(b) = [0.001; 0.999] \tag{3.214}$$

The total message node A receives from its parents nodes E and B is

$$\begin{aligned} \pi(a) &= \sum_{b,e} p(a|b,e)\pi_e(a)\pi_b(a) \\ &= \begin{bmatrix} \sum_{b,e} p(a=1|b,e)\pi_e(a)\pi_b(a) \\ \sum_{b,e} p(a=0|b,e)\pi_e(a)\pi_b(a) \end{bmatrix} = \begin{bmatrix} 0.002516 \\ 0 \end{bmatrix} \end{aligned} \tag{3.215}$$

where $p(a=0|b,e)=0$ as we revise given the evidence $a=1$. The message node A receives from its child J is

$$\begin{aligned} \lambda_j(a) &= \sum_j \lambda(j) \sum_{p_h} p(j|a,p_h)\pi_{p_h}(j) \\ &= \begin{bmatrix} \sum_j \lambda(j) \sum_{p_h} p(j|a=1,p_h)\pi_{p_h}(j) \\ \sum_j \lambda(j) \sum_{p_h} p(j|a=0,p_h)\pi_{p_h}(j) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \end{aligned} \tag{3.216}$$

where $\pi_{p_h}(j) = [1; 1]$ as initialized, and $p(j|a=0,p_h) = 0$ as we revise given the evidence $a=1$.

[1] Updating the messages and belief for node A is necessary as it is not a boundary evidence node and it's messages are needed to update the belief of other nodes.

The message node A receives from its child M is

$$\begin{aligned}\lambda_m(a) &= \sum_m \lambda(m)p(m|a) \\ &= \begin{bmatrix}\sum_m \lambda(m)p(m|a=1) \\ \sum_m \lambda(m)p(m|a=0)\end{bmatrix} = \begin{bmatrix}1 \\ 0\end{bmatrix}\end{aligned} \tag{3.217}$$

where $p(m|a=0)=0$ as we revise given the evidence $a=1$.

The total message node A receives from its children nodes J and M is

$$\lambda(a) = \lambda_j(a)\lambda_m(a) = \begin{bmatrix}2 \\ 0\end{bmatrix} \tag{3.218}$$

Given current messages, we obtain the normalized belief of node A:

$$Bel(a) = \alpha\pi(a)\lambda(a) = \begin{bmatrix}1 \\ 0\end{bmatrix} \tag{3.219}$$

Since node A is the evidence node, its belief doesn't change over iterations.

Messages computation and belief updating for node B

The message node B receives from its child A is

$$\begin{aligned}\lambda_a(b) &= \sum_a \lambda(a)\sum_e p(a|b,e)\pi_e(a) \\ &= \begin{bmatrix}\sum_a \lambda(a)\sum_e p(a|b=1,e)\pi_e(a) \\ \sum_a \lambda(a)\sum_e p(a|b=0,e)\pi_e(a)\end{bmatrix} = \begin{bmatrix}1.880040 \\ 0.003156\end{bmatrix}\end{aligned} \tag{3.220}$$

where $\pi_e(a) = [0.002; 0.998]$ is calculated in Eq. 3.213.

The total message node B receives from its child A is

$$\lambda(b) = \lambda_a(b) = \begin{bmatrix}1.880040 \\ 0.003156\end{bmatrix} \tag{3.221}$$

Given current messages, we obtain the normalized belief for node B

$$Bel(b) = \alpha\pi(b)\lambda(b) = \begin{bmatrix}0.373551 \\ 0.626449\end{bmatrix} \tag{3.222}$$

Message computation and belief updating for node E

The message node E receives from its child A is

$$\begin{aligned}\lambda_a(e) &= \sum_a \lambda(a)\sum_b p(a|b,e)\pi_b(a) \\ &= \begin{bmatrix}\sum_a \lambda(a)\sum_b p(a|b,e=1)\pi_b(a) \\ \sum_a \lambda(a)\sum_b p(a|b,e=0)\pi_b(a)\end{bmatrix} = \begin{bmatrix}0.581320 \\ 0.003878\end{bmatrix}\end{aligned} \tag{3.223}$$

where $\pi_b(a) = [0.001; 0.999]$ is calculated in Eq. 3.214.

The total message node E receives from its child A is

$$\lambda(e) = \lambda_a(e) = \begin{bmatrix} 0.581320 \\ 0.003878 \end{bmatrix} \tag{3.224}$$

Given current messages, we obtain the normalized belief of node E

$$Bel(e) = \alpha\pi(e)\lambda(e) = \begin{bmatrix} 0.231009 \\ 0.768991 \end{bmatrix} \tag{3.225}$$

Message computation and belief updating for node J

The message node J receives from its parent A is

$$\pi_a(j) = \pi(a) \prod_{c \in \text{Child}(a) \backslash j} \lambda_c(a) = \pi(a)\lambda_m(a) = \begin{bmatrix} 0.002516 \\ 0 \end{bmatrix} \tag{3.226}$$

where $\lambda_m(a) = [1; 0]$ is calculated in Eq. 3.217.

The message node J receives from its parent Ph is

$$\pi_{p_h}(j) = \pi(p_h) \prod_{c \in \text{Child}(p_h) \backslash j} \lambda_c(p_h) = \pi(p_h) = \begin{bmatrix} 0.05 \\ 0.95 \end{bmatrix} \tag{3.227}$$

The total message node J receives from its parent nodes Ph and A is

$$\begin{aligned} \pi(j) &= \sum_{p_h,a} p(j|p_h, a)\pi_{p_h}(j)\pi_a(j) \\ &= \begin{bmatrix} \sum_{p_h,a} p(j=1|p_h, a)\pi_{p_h}(j)\pi_a(j) \\ \sum_{p_h,a} p(j=0|p_h, a)\pi_{p_h}(j)\pi_a(j) \end{bmatrix} = \begin{bmatrix} 0.002271 \\ 0.000245 \end{bmatrix} \end{aligned} \tag{3.228}$$

where $\pi_a(j) = [0.002516; 0]$ is calculated in Eq. 3.226 and $\pi_{p_h}(j) = [0.05; 0.95]$ is calculated in Eq. 3.227.

Given current messages, we obtain the normalized belief for node J

$$Bel(j) = \alpha\pi(j)\lambda(j) = \begin{bmatrix} 0.902623 \\ 0.097377 \end{bmatrix} \tag{3.229}$$

Message computation and belief updating for node Ph

The message node Ph receives from its child J is

$$\begin{aligned} \lambda_j(p_h) &= \sum_j \lambda(j) \sum_a p(j|p_h, a)\pi_a(j) \\ &= \begin{bmatrix} \sum_j \lambda(j) \sum_a p(j|p_h=1, a)\pi_a(j) \\ \sum_j \lambda(j) \sum_a p(j|p_h=0, a)\pi_a(j) \end{bmatrix} = \begin{bmatrix} 0.002516 \\ 0.002516 \end{bmatrix} \end{aligned} \tag{3.230}$$

where $\pi_a(j) = [0.002516; 0]$ is calculated in Eq. 3.226.

The total message node Ph receives from child J is

$$\lambda(p_h) = \lambda_j(p_h) = \begin{bmatrix} 0.002516 \\ 0.002516 \end{bmatrix} \tag{3.231}$$

Given current messages, we obtain the normalized belief for node Ph

$$b_{p_h}(p_h) = \alpha\pi(p_h)\lambda(p_h) = \begin{bmatrix} 0.0500 \\ 0.9500 \end{bmatrix} \tag{3.232}$$

Message computation and belief updating for node M

The message node M receives from its parent A is

$$\pi_a(m) = \pi(a) \prod_{c \in \text{Child}(a) \setminus m} \lambda_c(a) = \pi(a)\lambda_j(a) = \begin{bmatrix} 0.005032 \\ 0 \end{bmatrix} \tag{3.233}$$

where $\lambda_j(a) = [2; 0]$ is calculated in Eq. 3.216.

The total message node M receives from its parent A is

$$\pi(m) = \sum_a p(m|a)\pi_a(m) = \begin{bmatrix} 0.003522 \\ 0.001510 \end{bmatrix} \tag{3.234}$$

Given current messages, we obtain the normalized belief of node M

$$Bel(m) = \alpha\pi(m)\lambda(m) = \begin{bmatrix} 0.7000 \\ 0.3000 \end{bmatrix} \tag{3.235}$$

We now finish the first iteration. We repeat the above process for the second and third iteration. Comparing the messages and beliefs from the second iteration and the third iteration, we can observe that there is no change. i.e., the belief propagation converges after the second iteration. In the end, we obtain the marginal distribution of each node given the evidence as

$$\begin{aligned} Bel(b) = p(b|a = 1) = [0.373551; 0.626449] \\ Bel(e) = p(e|a = 1) = [0.231009; 0.768991] \\ Bel(j) = p(j|a = 1) = [0.902623; 0.097377] \\ Bel(p_h) = p(p_h|a = 1) = [0.050000; 0.950000] \\ Bel(m) = p(m|a = 1) = [0.700000; 0.300000] \end{aligned} \tag{3.236}$$

3.9.6.2 Max-product inference given non-boundary evidence

To perform max-product inference, we only need to replace the summation operation with the maximization operation. The initialization remains the same. To illustrate the process, we perform the max-product inference given non-boundary evidence: `Alarm = 1`. The initialization, the revision of CPTs involving node A, and the order of nodes for updating remain the same as we did for sum-product inference given non-boundary evidence. In the following, we show the detailed calculation of messages and belief updating of each node for the first iteration.

Message computation and belief updating for node A

The message node A receives from its parent E is

$$\pi_e(a) = \pi(e) \prod_{c \in \text{Child}(e) \backslash a} \lambda_c(e) = \pi(e) = [0.002; 0.998] \tag{3.237}$$

The message node A receives from its parent B is

$$\pi_b(a) = \pi(b) \prod_{c \in \text{Child}(b) \backslash a} \lambda_c(b) = \pi(b) = [0.001; 0.999] \tag{3.238}$$

The total message node A receives from its parents nodes E and B is

$$\begin{aligned} \pi(a) &= \max_{b,e} p(a|b,e)\pi_e(a)\pi_b(a) \\ &= \begin{bmatrix} \max_{b,e} p(a=1|b,e)\pi_e(a)\pi_b(a) \\ \max_{b,e} p(a=0|b,e)\pi_e(a)\pi_b(a) \end{bmatrix} = \begin{bmatrix} 0.000997 \\ 0 \end{bmatrix} \end{aligned} \tag{3.239}$$

where $p(a=0|b,e)=0$ as we revise given the evidence $a=1$. The message node A receives from its child J is

$$\begin{aligned} \lambda_j(a) &= \max_j \lambda(j) \max_{p_h} p(j|a,p_h)\pi_{p_h}(j) \\ &= \begin{bmatrix} \max_j \lambda(j) \max_{p_h} p(j|a=1,p_h)\pi_{p_h}(j) \\ \max_j \lambda(j) \max_{p_h} p(j|a=0,p_h)\pi_{p_h}(j) \end{bmatrix} = \begin{bmatrix} 0.9500 \\ 0 \end{bmatrix} \end{aligned} \tag{3.240}$$

where $\pi_{p_h}(j) = [1;1]$ as initialized and $p(j|a=0,p_h)=0$ as we revise given the evidence $a=1$.

The message node A receives from its child M is

$$\begin{aligned} \lambda_m(a) &= \max_m \lambda(m)p(m|a) \\ &= \begin{bmatrix} \max_m \lambda(m)p(m|a=1) \\ \max_m \lambda(m)p(m|a=0) \end{bmatrix} = \begin{bmatrix} 0.7000 \\ 0 \end{bmatrix} \end{aligned} \tag{3.241}$$

where $p(m|a=0)=0$ as we revise given the evidence $a=1$.

The total message node A receives from its children nodes J and M is

$$\lambda(a) = \lambda_j(a)\lambda_m(a) = \begin{bmatrix} 0.665000 \\ 0 \end{bmatrix} \tag{3.242}$$

Given current messages, we obtain the normalized belief of node A:

$$Bel(a) = \alpha\pi(a)\lambda(a) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.243}$$

Since node A is the evidence node, its belief doesn't change over iterations.

Message computation and belief updating for node B

The message node B receives from its child A is

$$\begin{aligned} \lambda_a(b) &= \max_a \lambda(a) \max_e p(a|b,e)\pi_e(a) \\ &= \begin{bmatrix} \max_a \lambda(a) \max_e p(a|b=1,e)\pi_e(a) \\ \max_a \lambda(a) \max_e p(a|b=0,e)\pi_e(a) \end{bmatrix} = \begin{bmatrix} 0.623850 \\ 0.000664 \end{bmatrix} \end{aligned} \tag{3.244}$$

where $\pi_e(a) = [0.002; 0.998]$ is calculated in Eq. 3.237.

The total message node B receives from its child A is then

$$\lambda(b) = \lambda_a(b) = \begin{bmatrix} 0.623850 \\ 0.000664 \end{bmatrix} \tag{3.245}$$

Given current messages, we obtain the normalized belief of node B

$$Bel(b) = \alpha\pi(b)\lambda(b) = \begin{bmatrix} 0.484662 \\ 0.515338 \end{bmatrix} \tag{3.246}$$

Message computation and belief updating for node E

The message node E receives from its child A is

$$\begin{aligned} \lambda_a(e) &= \max_a \lambda(a) \max_b p(a|b,e)\pi_b(a) \\ &= \begin{bmatrix} \max_a \lambda(a) \max_b p(a|b,e=1)\pi_b(a) \\ \max_a \lambda(a) \max_b p(a|b,e=0)\pi_b(a) \end{bmatrix} = \begin{bmatrix} 0.192657 \\ 0.000664 \end{bmatrix} \end{aligned} \tag{3.247}$$

where $\pi_b(a) = [0.001; 0.999]$ is calculated in Eq. 3.238.

The total message node E receives from its child A is then

$$\lambda(e) = \lambda_a(e) = \begin{bmatrix} 0.192657 \\ 0.000664 \end{bmatrix} \tag{3.248}$$

Given current messages, we obtain the normalized belief of node E

$$Bel(e) = \alpha\pi(e)\lambda(e) = \begin{bmatrix} 0.367671 \\ 0.632329 \end{bmatrix} \tag{3.249}$$

Message computation and belief updating for node J

The message node J receives from its parent A is

$$\pi_a(j) = \pi(a) \prod_{c \in \text{Child}(a) \backslash j} \lambda_c(a) = \pi(a)\lambda_m(a) = \begin{bmatrix} 0.000698 \\ 0 \end{bmatrix} \tag{3.250}$$

where $\lambda_m(a) = [0.7000; 0]$ is calculated in Eq. 3.241.

The message node J receives from its parent Ph is

$$\pi_{p_h}(j) = \pi(p_h) \prod_{c \in \text{Child}(p_h) \backslash j} \lambda_c(p_h) = \pi(p_h) = \begin{bmatrix} 0.05 \\ 0.95 \end{bmatrix} \tag{3.251}$$

The total message node J receives from its parent nodes Ph and A is then

$$\begin{aligned} \pi(j) &= \max_{p_h, a} p(j|p_h, a)\pi_{p_h}(j)\pi_a(j) \\ &= \begin{bmatrix} \max_{p_h,a} p(j=1|p_h, a)\pi_{p_h}(j)\pi_a(j) \\ \max_{p_h,a} p(j=0|p_h, a)\pi_{p_h}(j)\pi_a(j) \end{bmatrix} = \begin{bmatrix} 0.000597 \\ 0.000066 \end{bmatrix} \end{aligned} \tag{3.252}$$

Given current messages, we obtain the normalized belief of node J

$$Bel(j) = \alpha\pi(j)\lambda(j) = \begin{bmatrix} 0.900452 \\ 0.099548 \end{bmatrix} \tag{3.253}$$

Message computation and belief updating for node Ph

The message node Ph receives from its child J is

$$\begin{aligned} \lambda_j(p_h) &= \max_j \lambda(j) \max_a p(j|p_h, a)\pi_a(j) \\ &= \begin{bmatrix} \max_j \lambda(j) \max_a p(j|p_h = 1, a)\pi_a(j) \\ \max_j \lambda(j) \max_a p(j|p_h = 0, a)\pi_a(j) \end{bmatrix} = \begin{bmatrix} 0.000663 \\ 0.000628 \end{bmatrix} \end{aligned} \tag{3.254}$$

where $\pi_a(j) = [0.000698; 0]$ is calculated in Eq. 3.250.

The total message node Ph receives from child J is then

$$\lambda(p_h) = \lambda_j(p_h) = \begin{bmatrix} 0.000663 \\ 0.000628 \end{bmatrix} \tag{3.255}$$

Given current messages, we obtain the normalized belief of node Ph

$$Bel(p_h) = \alpha\pi(p_h)\lambda(p_h) = \begin{bmatrix} 0.052640 \\ 0.947360 \end{bmatrix} \tag{3.256}$$

Message computation and belief updating for node M

The message node M receives from its parent A is

$$\pi_a(m) = \pi(a) \prod_{c \in \text{Child}(a) \setminus m} \lambda_c(a) = \pi(a)\lambda_j(a) = \begin{bmatrix} 0.000947 \\ 0 \end{bmatrix} \tag{3.257}$$

where $\lambda_j(a) = [0.9500; 0]$ is calculated in Eq. 3.240.

The total message node M receives from its parent A is then

$$\pi(m) = \max_a p(m|a)\pi_a(m) = \begin{bmatrix} 0.000663 \\ 0.000284 \end{bmatrix} \tag{3.258}$$

Given current messages, we obtain the normalized belief of node M

$$Bel(m) = \alpha\pi(m)\lambda(m) = \begin{bmatrix} 0.7000 \\ 0.3000 \end{bmatrix} \tag{3.259}$$

We now finish the first iteration. We repeat the above process for two more iterations, and observe that there is no change in the messages and beliefs at third iteration, i.e., the belief propagation converges after the second iteration. We can then find the unique MAP assignment by performing max marginal for each node independently if there are no ties in any of the updated node beliefs, yielding the MAP configuration for the example as

$$[0, 0, 1, 0, 1] = \arg \max_{b,e,j,p_h,m} p(b, e, j, p_h, m|a = 1) \tag{3.260}$$

via $x^* = \arg\max_x Bel(x)$ for $x \in \{B, E, J, Ph, M\}$.

3.9.7 EM learning for HMM

In this section, we introduce HMM learning using the standard EM method. Let an HMM be defined over the state variables $\mathbf{Q}$, observation variables $\mathbf{O}$, and parameters $\boldsymbol{\lambda} = \{\Lambda, A, B\}$. Given the training sequences $\mathbf{O} = \{O(m)\}_{m=1}^{M}$, where $O(m) = \{O^t(m)\}_{t=0}^{t_m}$, HMM learning is to find $\boldsymbol{\lambda}$ by maximizing it's log-likelihood, i.e.,

$$\boldsymbol{\lambda}^* = \arg \max_{\boldsymbol{\lambda}} \log p(\mathbf{O}|\boldsymbol{\lambda})$$

Let $Q(m)$ be the unobserved state sequence corresponding to $\mathbf{O}(m)$, $\log p(\mathbf{O}|\boldsymbol{\lambda})$ can be computed as follows

$$
\begin{aligned}
\log p(\mathbf{O}|\boldsymbol{\lambda}) &= \sum_{m=1}^{M} \log p(O(m)|\boldsymbol{\lambda}) \\
&= \sum_{m=1}^{M} \log \sum_{Q(m)} p(O(m), Q(m)|\boldsymbol{\lambda}), \\
&= \sum_{m=1}^{M} \log \sum_{Q(m)} q(Q(m)|O(m), \Theta_q) \frac{p(O(m), Q(m)|\boldsymbol{\lambda})}{q(Q(m)|O(m), \Theta_q)} \\
&\geq \sum_{m=1}^{M} \sum_{Q(m)} q(Q(m)|O(m), \Theta_q) \log p(O(m), Q(m)|\boldsymbol{\lambda}) \qquad \text{Jensen's inequality} \\
&= \sum_{m=1}^{M} \sum_{Q(m)} q(Q(m)|O(m), \Theta_q) \log p(O^0(m)|\Lambda) \prod_{t=1}^{t_m} p(O^t(m)|Q^t(m)|\mathbf{B}) p(Q^t(m)|Q^{t-1}(m), \mathbf{A}) \\
&= \sum_{m=1}^{M} \sum_{Q(m)} q(Q(m)|O(m), \Theta_q) \log p(O^0(m)|\Lambda) + \\
&\quad \sum_{m=1}^{M} \sum_{Q(m)} q(Q(m)|O(m), \Theta_q) \sum_{t=1}^{t_m} \log(O^t(m)|Q^t(m), \mathbf{B}) + \\
&\quad \sum_{m=1}^{M} \sum_{Q(m)} q(Q(m)|O(m), \Theta_q) \sum_{t=1}^{t_m} \log p(Q^t(m)|Q^{t-1}(m), \mathbf{A})
\end{aligned} \tag{3.261}
$$

It is clear from Eq. 3.261 that given $q(Q(m)|O(m), \Theta_q)$, parameters Λ, $\mathbf{A}$, and $\mathbf{B}$ can be computed separately by maximizing their expected likelihoods, corresponding to the three terms. Based on this, we can introduce the EM method as follows.

Set $q(Q(m)|O(m), \Theta_q) = p(Q(m)|O(m), \boldsymbol{\lambda}^{t-1})$ and initialize $\boldsymbol{\lambda}$ to $\boldsymbol{\lambda}^0$.
E-step:
Compute $p(Q(m)|O(m), \boldsymbol{\lambda}^{t-1})$ for all possible configurations of $Q(m)$ and for all sequences
M-step:
Find Λ^t, $\mathbf{A}^t$, and $\mathbf{B}^t$ by maximizing the their expected loglikelihood, i.e.,

$$
\begin{aligned}
\Lambda^t &= \arg\max_{\Lambda} \sum_{m=1}^{M} \sum_{Q(m)} p(Q(m)|O(m), \boldsymbol{\lambda}^{t-1}) \log p(O^0(m)|\Lambda) \\
\mathbf{B}^t &= \arg\max_{\mathbf{B}} \sum_{m=1}^{M} \sum_{Q(m)} p(Q(m)|O(m), \boldsymbol{\lambda}^{t-1}) \sum_{t=1}^{t_m} \log(O^t(m)|Q^t(m), \mathbf{B}) \\
\mathbf{A}^t &= \arg\max_{\mathbf{A}} \sum_{m=1}^{M} \sum_{Q(m)} p(Q(m)|O(m), \boldsymbol{\lambda}^{t-1}) \sum_{t=1}^{t_m} \log p(Q^t(m)|Q^{t-1}(m), \mathbf{A})
\end{aligned}
$$

Repeat the E and M steps until convergence

Algorithm 3.16 HMM EM learning pseudo-code

```
Initialize $\boldsymbol{\lambda}$ to $\boldsymbol{\lambda}^0$ and $w(c,m)$ and $S(m,t,i,j)$ to zeros
E-step:
for $m = 1$ to $M$ do
    for $c = 1$ to $C_m$ do                ▷ $C_m$ is the number of configurations of $Q(m)$
        $w(c,m) = p(Q_c(m), O(m)|\boldsymbol{\lambda}^{t-1})$ ▷ $Q_c(m)$ is the $c$-th configuration of $Q(m)$
    end for
end for
Compute the expected state transition counts
for $m = 1$ to $M$ do
    for $c = 1$ to $C_m$ do
        for $i = 1$ to $N$ do
            for $j = 1$ to $N$ do
                $S(m,t,i,j) = S(m,t,i,j) + I(Q_c^{t-1}(m) = i \wedge Q_c^t(m) = j)w(c,m)$
            end for
        end for
    end for
end for
M-step:
    $a_{ij} = \frac{\sum_{m=1}^{M}\sum_{t=0}^{t_m} S(m,t,i,j)}{\sum_{m=1}^{M}\sum_{t=1}^{t_m}\sum_{j=1}^{N} S(m,t,i,j)}$
    $b_i(k) = \frac{\sum_{m=1}^{M}\sum_{t=0}^{t_m}\sum_{j=1}^{N} S(m,t,i,j)I(O^t(m) = k)}{\sum_{m=1}^{M}\sum_{t=0}^{t_m}\sum_{j=1}^{N} S(m,t,i,j)}$
    $\pi_i = \frac{\sum_{m=1}^{M}\sum_{j=1}^{N} S(m,0,i,j)}{\sum_{m=1}^{M} C_m}$
Repeat E and M step until convergence
```

3.9.8 Discrete BN structure learning with marginal likelihood score

Given the training data, $\mathcal{D} = \{D_1, D_2, ..., D_M\}$, where $D_m = \{x_1^m, x_2^m, ..., x_N^m\}$, the maximum likelihood learning of the BN structure can be formulated as finding the BN structure that maximizes the marginal log likelihood of the structure $\mathcal{G}$, i.e.,

$$\mathcal{G}^* = \arg\max_{\mathcal{G}} \log p(\mathcal{D}|\mathcal{G}) \tag{3.262}$$

where $\log p(\mathcal{D}|\mathcal{G})$ is the log marginal likelihood of $\mathcal{G}$ given the training data $\mathcal{D}$. As $\log p(\mathcal{D}|\mathcal{G})$ is decomposable, it can be rewritten as

$$\log p(\mathcal{D}|\mathcal{G}) = \log \prod_{n=1}^{N} p(\mathcal{D}|G_n) = \sum_{n=1}^{N} \log p(\mathcal{D}|G_n) \tag{3.263}$$

where $\mathcal{G}_n = \{x_n, \pi(x_n)\}$ represents the local structure for node x_n. Hence, the structure for each node can be learned separately, i.e.,

$$\mathcal{G}_n^* = \arg\max_{\mathcal{G}_n} \log p(\mathcal{D}|\mathcal{G}_n) \tag{3.264}$$

By conditioning on θ_n, the parameters for node n, $\log p(\mathcal{D}|\mathcal{G}_n)$ can be rewritten as

$$\log p(\mathcal{D}|\mathcal{G}_n) = \log \int p(\mathcal{D}|\mathcal{G}_n, \theta_n) p(\theta_n|G_n) d\theta_n \tag{3.265}$$

where the first term is the joint likelihood of the structure and parameters for node x_n and the second term is the prior distribution of x_n's parameters given its structure. For a discrete BN, where $x_n \in \{1, 2, ...K_n\}$, θ_n can be written as $\theta_n = \{\theta_{nj}\}_{j=1}^{J_n}$, where j is the index to the jth parent configuration and J_n is the total number of parent configurations for node x_n. Assuming θ_{nj} are independent of each other, $p(\theta_n|\mathcal{G}_n)$ can be calculated as

$$p(\theta_n|\mathcal{G}_n) = \prod_{j=1}^{J_n} p(\theta_{nj}|\mathcal{G}_n) = \prod_{j=1}^{J_n} \frac{\prod_{k=1}^{K_n} \theta_{njk}^{\alpha_{njk}-1}}{B(\alpha_{njk})} \tag{3.266}$$

where θ_{njk} is assumed to follow Dirichlet distribution with hyper-parameters α_{njk}, and $B(\alpha_{njk}) = \frac{\prod_{k=1}^{K_n} \Gamma(\alpha_{njk})}{\Gamma(\sum_{k=1}^{K_n} \alpha_{njk})}$ and $\Gamma()$ is the Gamma function.

Furthermore,

$$\begin{aligned} p(\mathcal{D}|\mathcal{G}_n, \theta_n) &= \prod_{m=1}^{M} p(D_m|\mathcal{G}_n, \theta_n) \\ &= \prod_{m=1}^{M} p(x_n^m|\pi(x_n)^m, \theta_n) = \prod_{m=1}^{M} \prod_{j=1}^{J_n} p(x_n^m|\pi(x_n)^m = j, \theta_{nj}) \\ &= \prod_{m=1}^{M} \prod_{j=1}^{J_n} \prod_{k=1}^{K_n} \theta_{njk}^{I(x_n^m=k\&\pi(x_n)^m=j}) = \prod_{j=1}^{J_n} \prod_{k=1}^{K_n} \theta_{njk}^{\sum_{m=1}^{M} I(x_n^m=k\&\pi(x_n)^m=j}) \\ &= \prod_{j=1}^{J_n} \prod_{k=1}^{K_n} \theta^{N_{njk}} \end{aligned} \tag{3.267}$$

where $I()$ is the indicator function and N_{njk} is the counts for node x_n with a value of k and jth parent configuration.

Incorporating Eq. 3.266 and Eq. 3.267 into Eq. 3.265 yields

$$
\begin{aligned}
\log p(\mathcal{D}|\mathcal{G}_n) &= \log \int p(\mathcal{D}|\mathcal{G}_n, \theta_n) p(\theta_n|G_n) \\
&= \log \int \prod_{j=1}^{J_n} \prod_{k=1}^{K_n} \theta^{N_{njk}} \prod_{j=1}^{J_n} \frac{\prod_{k=1}^{K_n} \theta_{njk}^{\alpha_{njk}-1}}{B(\alpha_{njk})} d\theta_{njk} \\
&= \log \frac{\prod_{j=1}^{J_n} \int \prod_{k=1}^{K_n} \theta^{N_{njk}+\alpha_{njk}-1}}{B(\alpha)} d\theta_{njk} \\
&= \sum_{j=1}^{J_n} \log \frac{B(\alpha_{njk}+N_{njk})}{B(\alpha_{njk})} \underbrace{\int \frac{\prod_{k=1}^{K_n} \theta^{N_{njk}+\alpha_{njk}-1}}{B(\alpha_{njk}+N_{njk})} d\theta_{njk}}_{\text{integration to 1}} \\
&= \sum_{j=1}^{J_n} \log \frac{B(\alpha_{njk}+N_{njk})}{B(\alpha_{njk})} \qquad (3.268)
\end{aligned}
$$

Incorporating Eq. 3.268 into Eq. 3.263 yields

$$
\begin{aligned}
\log p(\mathcal{D}|\mathcal{G}) &= \sum_{n=1}^{N} \sum_{j=1}^{J_n} \log \frac{B(\alpha_{njk}+N_{njk})}{B(\alpha_{njk})} \\
&= \sum_{n=1}^{N} \sum_{j=1}^{J_n} \log \frac{\Gamma(\sum_{k=1}^{K_n} \alpha_{njk}) \prod_{k=1}^{K_n} \Gamma(N_{njk}+\alpha_{njk})}{\prod_{k=1}^{K_n} \Gamma(\alpha_{njk}) \Gamma(\sum_{k=1}^{K_n}(N_{njk}+\alpha_{njk}))} \qquad (3.269)
\end{aligned}
$$

Assuming symmetric prior, i.e., $\alpha_{njk} = \frac{\alpha}{K_n \times J_n}$, where α is a tuning parameter called equivalent sample size, Eq. 3.263 can be rewritten as

$$
\log p(\mathcal{D}|\mathcal{G}) = \sum_{n=1}^{N} \sum_{j=1}^{J_n} \log \frac{\Gamma(\alpha_{njk} K_n) \prod_{k=1}^{K_n} \Gamma(N_{njk}+\alpha_{njk})}{\Gamma^{K_n}(\alpha_{njk}) \Gamma(N_{nj}+\alpha_{njk} K_n)} \qquad (3.270)
$$

Further assuming uniform prior, i.e., $\alpha_{njk} = 1$, Eq. 3.269 can be rewritten as

$$
\log p(\mathcal{D}|\mathcal{G}) = \sum_{n=1}^{N} \sum_{j=1}^{J_n} \log \frac{(K_n-1)! \prod_{k=1}^{K_n} N_{njk}!}{(N_{nj}+K_n-1)!} \qquad (3.271)
$$

where we use the fact that $\Gamma(1) = 1$ and $\Gamma(n+1) = n!$ for $n > 0$. Alternative independent derivations of the marginal likelihood score may be found in [11]. The marginal likelihood score is also called Bayesian Dirichlet score in the literature. And Eq. 3.270 is called the Bayesian Dirichlet Equivalent Uniform (BDeu) score and Eq. 3.271 is the K2 score. Compared to the BIC score, the marginal likelihood score does not have any assumptions such as a large number of samples and the Gaussian distribution, and empirical results show that it outperforms BIC score in the accuracy with the learned BN structures, in particular when the amount of training data is small. Additional analysis on the marginal likelihood score for learning discrete BN structures may be found in [12].

4.8 Chapter 4 Appendix

4.8.1 Examples for Belief Propagation in MRF

Given the the binary MRF in Figure 4.10,

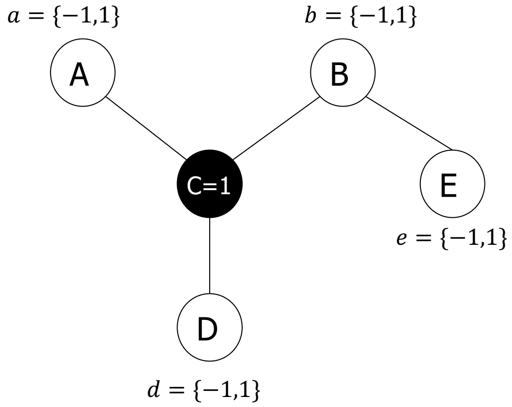

Figure 4.10: Structure of a binary MRF with evidence

it's unary potentials,

$$\begin{aligned}
\phi_A(a) &= \begin{bmatrix} \exp(-1.2) \\ \exp(2) \end{bmatrix} \\
\phi_B(b) &= \begin{bmatrix} \exp(0.8) \\ \exp(-0.2) \end{bmatrix} \\
\phi_C(c) &= \begin{bmatrix} \exp(-1.3) \\ \exp(-0.2) \end{bmatrix} \\
\phi_D(d) &= \begin{bmatrix} \exp(0.2) \\ \exp(-0.2) \end{bmatrix} \\
\phi_E(e) &= \begin{bmatrix} \exp(-0.5) \\ \exp(0.5) \end{bmatrix}
\end{aligned} \tag{4.67}$$

and pairwise potentials,

$$\begin{aligned}
\psi_{AC}(a,c) &= \begin{bmatrix} \exp(2) & \exp(-1) \\ \exp(-1) & \exp(2) \end{bmatrix} \\
\psi_{BC}(b,c) &= \begin{bmatrix} \exp(-0.3) & \exp(1.2) \\ \exp(1.2) & \exp(-0.3) \end{bmatrix} \\
\psi_{BE}(b,e) &= \begin{bmatrix} \exp(0.5) & \exp(1) \\ \exp(1) & \exp(0.5) \end{bmatrix} \\
\psi_{DC}(d,c) &= \begin{bmatrix} \exp(0.9) & \exp(-0.2) \\ \exp(-0.2) & \exp(0.9) \end{bmatrix}
\end{aligned} \tag{4.68}$$

we show below the process for performing belief propagation for both sum-product and max-product inference for this MRF.

4.8.1.1 Sum-product inference with evidence

We now perform sum-product inference, given $c = 1$. During initialization, the entries of the unary and pairwise potential functions involving the evidence node C corresponding to $c = 0$ are set to zero and remain unchanged otherwise, i.e.,

$$\begin{aligned}
\phi_C(c) &= \begin{bmatrix} 0 \\ \exp(-0.2) \end{bmatrix} \\
\psi_{AC}(a,c) &= \begin{bmatrix} 0 & \exp(-1) \\ 0 & \exp(2) \end{bmatrix} \\
\psi_{BC}(b,c) &= \begin{bmatrix} 0 & \exp(1.2) \\ 0 & \exp(-0.3) \end{bmatrix} \\
\psi_{DC}(d,c) &= \begin{bmatrix} 0 & \exp(-0.2) \\ 0 & \exp(0.9) \end{bmatrix}
\end{aligned} \tag{4.69}$$

Messages for all nodes are initialized to ones. For each node, we update the messages it receives from its neighbors based on their current messages. We calculate messages and belief of each node using Eq. (4.30) and (4.31) as shown below.

Node A: The message node A receives from its neighbor node C is calculated as

$$m_{CA}(a) = \sum_c \phi_C(c)\psi_{AC}(a,c)m_{BC}(c)m_{DC}(c) = \begin{bmatrix} 0.3012 \\ 6.0496 \end{bmatrix} \tag{4.70}$$

where m_{BC} and m_{DC} take on their current values (i.e., initial values). The belief of node A given current message is then

$$Bel(a) = \alpha\phi_A(a)m_{CA}(a) = \begin{bmatrix} 0.0020 \\ 0.9980 \end{bmatrix} \tag{4.71}$$

where α is the normalization constant.

Node B: The messages node B receives from its neighbor nodes C and E are calculated as

$$m_{CB}(b) = \sum_c \phi_C(c)\psi_{BC}(b,c)m_{AC}(c)m_{DC}(c) = \begin{bmatrix} 2.7183 \\ 0.6065 \end{bmatrix} \tag{4.72}$$

$$m_{EB}(b) = \sum_e \phi_E(e)\psi_{BE}(b,e) = \begin{bmatrix} 5.4817 \\ 4.3670 \end{bmatrix} \tag{4.73}$$

where m_{AC} and m_{DC} assume their current values. The belief of node B given current messages is then

$$Bel(b) = \alpha\phi_B(b)m_{CB}(b)m_{EB}(b) = \begin{bmatrix} 0.9386 \\ 0.0614 \end{bmatrix} \tag{4.74}$$

Node C: The messages node C receives from its neighbor nodes A, B and, D are calculated as

$$m_{AC}(c) = \sum_a \phi_A(a)\psi_{AC}(a,c) = \begin{bmatrix} 0 \\ 54.7090 \end{bmatrix} \tag{4.75}$$

$$m_{BC}(c) = \sum_b \phi_B(b)\psi_{BC}(b,c)m_{EB}(b) = \begin{bmatrix} 0 \\ 43.1532 \end{bmatrix} \tag{4.76}$$

$$m_{DC}(c) = \sum_d \phi_D(d)\psi_{DC}(d,c) = \begin{bmatrix} 0 \\ 3.0138 \end{bmatrix} \tag{4.77}$$

where $m_{EB} = [5.4817; 4.3670]$ is calculated in Eq. 4.73. The belief of node C given current messages is then

$$Bel(c) = \alpha\phi_C(c)m_{AC}(c)m_{BC}(c)m_{DC}(c) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.78}$$

Since node C is the evidence node with $c = 1$, its belief doesn't change over iterations.

Node D: The message node D receives from its neighbor node C is calculated as

$$m_{CD}(d) = \sum_c \phi_C(c)\psi_{CD}(c,d)m_{AC}(c)m_{BC}(c) = \begin{bmatrix} 1582.5 \\ 4754.2 \end{bmatrix} \tag{4.79}$$

where $m_{AC} = [0; 54.7090]$ is calculated in Eq. 4.75 and $m_{BC} = [0; 43.1532]$ is calculated in Eq. 4.76. The belief of node D given current message is then

$$Bel(d) = \alpha\phi_D(d)m_{CD}(d) = \begin{bmatrix} 0.3318 \\ 0.6682 \end{bmatrix} \tag{4.80}$$

Node E: The message node E receives from its neighbor node B is calculated as

$$m_{BE}(e) = \sum_{b} \phi_B(b)\psi_{BE}(b,e)m_{CB}(b) = \begin{bmatrix} 11.3240 \\ 17.2634 \end{bmatrix} \tag{4.81}$$

where $m_{CB} = [2.7183; 0.6065]$ is calculated in Eq. 4.72. The belief of node E given current message is then

$$Bel(e) = \alpha\phi_E(e)m_{BE}(e) = \begin{bmatrix} 0.1944 \\ 0.8056 \end{bmatrix} \tag{4.82}$$

We now finish the first iteration. We repeat the process for several times and observe that the messages do not change in the third iteration. Thus, the belief propagation converges after the second iteration. In the end, we obtain the belief of each node given the evidence as

$$\begin{aligned} Bel(a) &= p(a|c=1) = [0.0020; 0.9980] \\ Bel(b) &= p(b|c=1) = [0.9386; 0.0614] \\ Bel(d) &= p(d|c=1) = [0.3318; 0.6682] \\ Bel(e) &= p(e|c=1) = [0.1944; 0.8056] \end{aligned} \tag{4.83}$$

4.8.1.2 Max-product inference with evidence

To perform max-product inference, we only need to replace the summation operation in the sum-product inference with the maximization operation, and the remaining calculations remain the same. We follow the same procedure to initialize messages for all nodes to ones. Unary and pairwise potential functions corresponding to the unobserved state of the evidence node are set to 0, and remain unchanged otherwise. Max-product propagation can then start.

Node A: The message node A receives from its neighbor node C is calculated as

$$m_{CA}(a) = \max_{c} \phi_C(c)\psi_{AC}(a,c)m_{BC}(c)m_{DC}(c) = \begin{bmatrix} 0.3012 \\ 6.0496 \end{bmatrix} \tag{4.84}$$

where m_{BC} and m_{DC} are all ones as initialized. The belief of node A given current message is then

$$Bel(a) = \alpha\phi_A(a)m_{CA}(a) = \begin{bmatrix} 0.0020 \\ 0.9980 \end{bmatrix} \tag{4.85}$$

Node B: The messages node B receives from its neighbor nodes C and E are calculated as

$$m_{CB}(b) = \max_{c} \phi_C(c)\psi_{BC}(b,c)m_{AC}(c)m_{DC}(c) = \begin{bmatrix} 2.7183 \\ 0.6065 \end{bmatrix} \tag{4.86}$$

$$m_{EB}(b) = \max_{e} \phi_E(e)\psi_{BE}(b,e) = \begin{bmatrix} 4.4817 \\ 2.7183 \end{bmatrix} \tag{4.87}$$

where m_{AC} and m_{DC} are all ones as initialized. The belief of node B given current messages is then

$$Bel(b) = \alpha\phi_B(b)m_{CB}(b)m_{EB}(b) = \begin{bmatrix} 0.9526 \\ 0.0474 \end{bmatrix} \tag{4.88}$$

Node C: The messages node C receives from its neighbor nodes A, B, and D are calculated as

$$m_{AC}(c) = \max_a \phi_A(a)\psi_{AC}(a,c) = \begin{bmatrix} 0 \\ 54.5982 \end{bmatrix} \tag{4.89}$$

$$m_{BC}(c) = \max_b \phi_B(b)\psi_{BC}(b,c)m_{EB}(b) = \begin{bmatrix} 0 \\ 33.1155 \end{bmatrix} \tag{4.90}$$

$$m_{DC}(c) = \max_d \phi_D(d)\psi_{DC}(d,c) = \begin{bmatrix} 0 \\ 2.0138 \end{bmatrix} \tag{4.91}$$

where $m_{EB} = [4.4817; 2.7183]$ is calculated in Eq. 4.87. The belief of node C given current messages is then

$$Bel(c) = \alpha\phi_C(c)m_{AC}(c)m_{BC}(c)m_{DC}(c) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.92}$$

Since node C is the evidence node with $c = 1$, its belief doesn't change over iterations.

Node D: The message node D receives from its neighbor node C is calculated as

$$m_{CD}(d) = \max_c \phi_C(c)\psi_{CD}(c,d)m_{AC}(c)m_{BC}(c) = \begin{bmatrix} 1212.0 \\ 3641.0 \end{bmatrix} \tag{4.93}$$

where $m_{AC} = [0; 54.5982]$ and $m_{BC} = [0; 33.1155]$ are calculated in Eq. 4.89 and Eq. 4.90 respectively. The belief of node D given current message is then

$$Bel(d) = \alpha\phi_D(d)m_{CD}(d) = \begin{bmatrix} 0.3318 \\ 0.6682 \end{bmatrix} \tag{4.94}$$

Node E: The message node E receives from its neighbor node B is calculated as

$$m_{BE}(e) = \max_b \phi_B(b)\psi_{BE}(b,e)m_{CB}(b) = \begin{bmatrix} 9.9742 \\ 16.4446 \end{bmatrix} \tag{4.95}$$

where $m_{CB} = [2.7183; 0.6065]$ is calculated in Eq. 4.86. The belief of node E given current message is then

$$Bel(e) = \alpha\phi_E(e)m_{BE}(e) = \begin{bmatrix} 0.1824 \\ 0.8176 \end{bmatrix} \tag{4.96}$$

We now finish the first iteration. We repeat the process for several iterations and observe that the messages do not change in the third iteration. Thus, the belief propagation converges after the second iteration. Given the beliefs, we have the MAP configuration as

$$[1, -1, 1, 1] = \arg\max_{a,b,d,e} p(a, b, d, e|c = 1) \tag{4.97}$$

where $x^* = \arg\max_x Bel(x)$ for $x \in \{A, B, D, E\}$.

References

[1] R. C. Jeffrey, The logic of decision. University of Chicago Press, 1990.

[2] J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier, 2014.

[3] H. Chan and A. Darwiche, "On the revision of probabilistic beliefs using uncertain evidence," Artificial Intelligence, vol. 163, no. 1, pp. 67–90, 2005.

[4] M. Valtorta, Y.-G. Kim, and J. Vomlel, "Soft evidential update for probabilistic multiagent systems," International Journal of Approximate Reasoning, vol. 29, no. 1, pp. 71–106, 2002.

[5] Y. Peng, S. Zhang, and R. Pan, "Bayesian network reasoning with uncertain evidences," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 18, no. 05, pp. 539–564, 2010.

[6] J. Bilmes, "On virtual evidence and soft evidence in bayesian networks," 2004.

[7] F. J. Groen and A. Mosleh, "Foundations of probabilistic inference with uncertain evidence," International Journal of Approximate Reasoning, vol. 39, no. 1, pp. 49–83, 2005.

[8] R. Pan, Y. Peng, and Z. Ding, "Belief update in bayesian networks using uncertain evidence," in 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), pp. 441–444, IEEE, 2006.

[9] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid monte carlo," Physics letters B, vol. 195, no. 2, pp. 216–222, 1987.

[10] R. M. Neal et al., "Mcmc using hamiltonian dynamics," Handbook of markov chain monte carlo, vol. 2, no. 11, p. 2, 2011.

[11] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," Machine learning, vol. 20, no. 3, pp. 197–243, 1995.

[12] M. Scutari, "An empirical-bayes score for discrete bayesian networks," in Conference on probabilistic graphical models, pp. 438–448, PMLR, 2016.