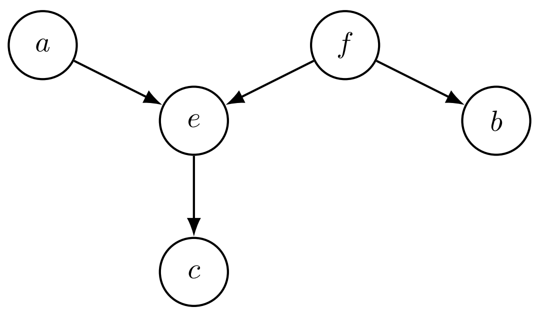
Tutorial: Probabilistic Graphical Models

Independence, Naive Bayes, and Variable Elimination

1 Exercise 1: Conditional Independence (Pruning / Edge Deletion)

Consider the following Directed Acyclic Graph (DAG):



1. Joint Distribution Factorization

$$p(a, b, c, e, f) = p(a)p(f)p(e|a, f)p(b|f)p(c|e)$$

2. Verification of Independence

We apply the **Pruning / Edge Deletion** algorithm. To test $\mathbf{X} \perp \mathbf{Z} \mid \mathbf{Y}$:

1. **Delete Barren Nodes:** Recursively delete leaf nodes not in $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$.
2. **Delete Outgoing Edges:** Remove all edges originating from nodes in the conditioning set $\mathbf{Y}$.
3. **Check Connectivity:** If $\mathbf{X}$ and $\mathbf{Z}$ are disconnected in the undirected skeleton of the resulting graph, they are independent.

Case (a): Is $a \perp b \,|\, c$?

Step 1: Delete Nodes

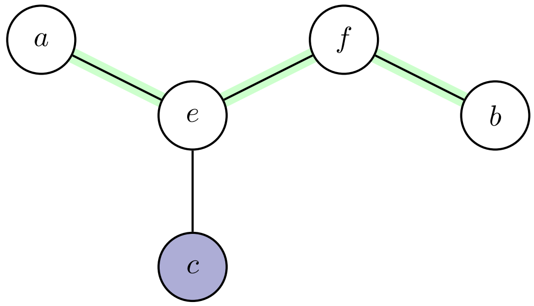
Target set is $\{a, b, c\}$.

- Leaves in G: c and b. Both are in the target set.
- **Result:** No nodes are deleted.

Step 2: Delete Outgoing Edges from Conditioning Set $\{c\}$

- Node c has no outgoing edges.
- **Result:** No edges are deleted.

Step 3: Check Connectivity In the remaining graph (which is identical to the original), is there a path between a and b?



Conclusion: There is a path $a - e - f - b$ in the undirected skeleton.

$$a \not\perp b \mid c$$

Case (b): Is $a \perp b \mid f$?

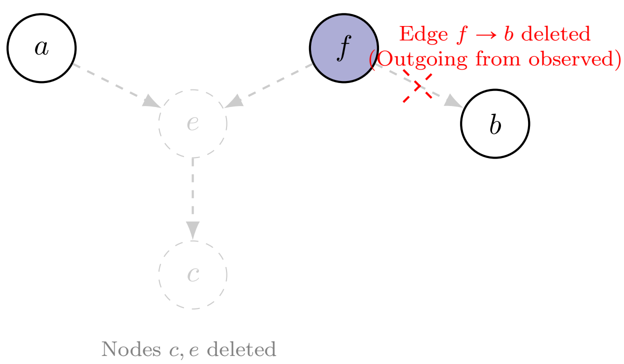
Step 1: Delete Barren Nodes

Target set is $\{a, b, f\}$.

- Node c is a leaf and $c \notin \{a, b, f\}$. **Delete c.**
- After deleting c, node e becomes a leaf.
- Node e is a leaf and $e \notin \{a, b, f\}$. **Delete e.**
- Node a is now a leaf, but $a \in$ Target. Keep.
- Node b is a leaf, but $b \in$ Target. Keep.

Step 2: Delete Outgoing Edges from Conditioning Set $\{f\}$

- Edges starting at f: The edge $f \to b$ exists.
- **Delete $f \to b$.** (Note: $f \to e$ was already removed when we deleted node e).



Step 3: Check Connectivity In the final graph, node a is isolated and node b is isolated. There is no path connecting them.

$$a \perp b \mid f$$

2 Exercise 2: Naive Bayes Model

1. Problem Setting

Consider the inference problem of text classification into **spam** ($C = 1$) or **not spam** ($C = 0$).

Bag of Words Representation: Suppose we have a dictionary of D words $\mathcal{D} = \{W_1, \ldots, W_D\}$ as an indexable set. A text x is a set of words in the dictionary, i.e., $x = \{W \in \mathcal{D}\}$, which can equivalently be represented as a set of indices $x' = \{i : W_i \in x\}$.

Note: This is a fancy way of saying "appearance of word matters, repetition and order doesn't matter".

Example: Let $\mathcal{D} = \{\text{hello, world, test, is, this, a}\}$ with $D = 6$.

- "hello world" $\equiv$ {hello, world} $\equiv \{1, 2\}$
- "this is a test" $\equiv$ {test, is, this, a} $\equiv \{3, 4, 5, 6\}$
- "hello hello hello world" $\equiv \{1, 2\} =$ "hello world" $=$ "world hello"

Let $X = (X_1, \ldots, X_D)$ where $X_i \in \{0, 1\}$ is a binary random vector denoting the appearance of the i-th word in the text (e.g., $X(\text{hello world}) = (1, 1, 0, 0, 0, 0)$). Our goal is to compute the posterior $p(C|X)$.

2. A General Model

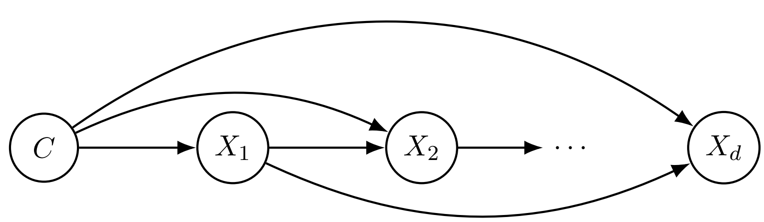
Using Bayes' theorem, we can write the posterior as:

$$p(C|X) = \frac{p(C, X)}{p(X)}$$

Since the denominator $p(X)$ does not depend on the specific outcome of C, we have $p(C|X) \propto p(C, X)$. We can factorize $p(C, X)$ into its components using the chain rule:

$$\begin{aligned} p(C, X) &= p(C)p(X|C) \\ &= p(C)p(X_1|C)p(X_2|X_1, C) \ldots p(X_d|X_1, \ldots, X_{d-1}, C) \\ &= p(C)p(X_1|C) \prod_{i=2}^{d} p(X_i|X_1, \ldots, X_{i-1}, C) \end{aligned}$$

Graphical Model (General): Since each term is conditioned on all variables that appeared to its left, the Directed Graphical Model (DGM) is fully connected:



Observations on Complexity:

- This graph has $d + 1$ nodes (X_1 to X_d, and C).
- The graph is fully connected; every node is a neighbor of every other node.
- For node X_i, the number of input edges is i (neighbors $C, X_1, \dots, X_{i-1}$).
- The size of the conditional probability table (CPT) for each node requires $2^{\#\text{input edges}}$ parameters.
- **Total # of parameters:**

$$1 + \sum_{i=1}^{d} 2^i = 1 + (2^{d+1} - 2) = 2^{d+1} - 1$$

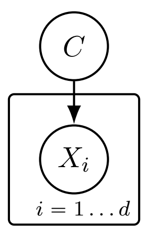
  This equals the number of parameters needed to specify the joint tensor over $d + 1$ binary random variables. The complexity scales exponentially.

3. Reducing Complexity with Naive Bayes

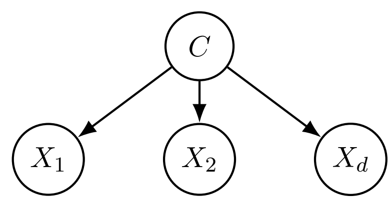
Learning $2^{d+1} - 1$ parameters is very expensive (computationally and learning-theoretically).

Goal: Reduce parameters by simplifying the graphical model.

Method: Remove all edges between (X_i, X_j); only keep edges originating from C.



Naive Bayes (Plate Notation)



Naive Bayes (Explicit)

Implied Factorization:

$$p(X, C) = p(C) \prod_{i=1}^{d} p(X_i|C)$$

This implies $p(X_i|X_1, \dots, X_{i-1}, C) = p(X_i|C)$. In other words, X_i is independent from X_j for all $j \neq i$ given C.

Conclusion:

- We can manipulate the joint distribution through manipulating the DGM!
- **Number of parameters:** $1+2d$. The complexity now scales linearly instead of exponentially.

3 Exercise 3: Gaussian Log-Likelihood

Gaussian log-likelihood

Suppose we observe some i.i.d. data $\mathbf{x}_{1:n} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ from the m-variate Gaussian distribution $\mathcal{N}_m(\mu, \Sigma)$. The density is:

$$f(\mathbf{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{m/2}} (\det \Sigma)^{-1/2} \exp\{-\tfrac{1}{2}(\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu)\}.$$

It is convenient to equivalently express this density in terms of $K = \Sigma^{-1}$:

$$f(\mathbf{x}; \mu, K) = \frac{1}{(2\pi)^{m/2}} (\det K)^{1/2} \exp\{-\tfrac{1}{2}(\mathbf{x} - \mu)^\top K (\mathbf{x} - \mu)\},$$

after taking logarithms it becomes

$$\log f(\mathbf{x}; \mu, K) = -\frac{m}{2} \log(2\pi) + \frac{1}{2} \log \det K - \frac{1}{2} (\mathbf{x} - \mu)^\top K (\mathbf{x} - \mu).$$

Up to the obvious constants that do not depend on μ and K, the log-likelihood is

$$\ell_n(\mu, K) = \sum_{i=1}^n \log f(\mathbf{x}_i; \mu, K) = (\text{const}) + \frac{n}{2} \log \det(K) - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \mu)^\top K (\mathbf{x}_i - \mu).$$

MLE for the Mean μ

Irrespective of the value of K, the optimal $\hat{\mu}$ satisfies

$$\hat{\mu} = \bar{\mathbf{x}}_n = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$

This is because the gradient of $\nabla_\mu \ell_n$ is

$$\nabla_\mu \ell_n(\mu, K) = -\frac{1}{2} \sum_{i=1}^n (2K\mu - 2K\mathbf{x}_i) = -nK\mu + K \sum_{i=1}^n \mathbf{x}_i = nK(\bar{\mathbf{x}}_n - \mu).$$

Since K is invertible, this can be zero if and only if $\mu = \bar{\mathbf{x}}_n$.

Profile Likelihood for K

We can thus consider the profile likelihood

$$\ell_n(\bar{\mathbf{x}}_n, K) = (\text{const}) + \frac{n}{2} \log \det(K) - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}_n)^\top K (\mathbf{x}_i - \bar{\mathbf{x}}_n).$$

Note that

$$\begin{aligned}
\sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}_n)^\top K (\mathbf{x}_i - \bar{\mathbf{x}}_n) &= \sum_{i=1}^n \operatorname{tr}((\mathbf{x}_i - \bar{\mathbf{x}}_n)^\top K (\mathbf{x}_i - \bar{\mathbf{x}}_n)) \\
&= \sum_{i=1}^n \operatorname{tr}(K (\mathbf{x}_i - \bar{\mathbf{x}}_n)(\mathbf{x}_i - \bar{\mathbf{x}}_n)^\top) \\
&= n \operatorname{tr}\left(K \left\{ \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}_n)(\mathbf{x}_i - \bar{\mathbf{x}}_n)^\top \right\} \right) \\
&= n \operatorname{tr}(K S_n),
\end{aligned}$$

where S_n is the sample covariance matrix. Note that $\bar{\mathbf{x}}_n$ and S_n form the sufficient statistics for the Gaussian model. With this new notation:

$$\ell_n(\bar{\mathbf{x}}_n, K) = (\text{const}) + \frac{n}{2}(\log\det(K) - \text{tr}(KS_n)).$$

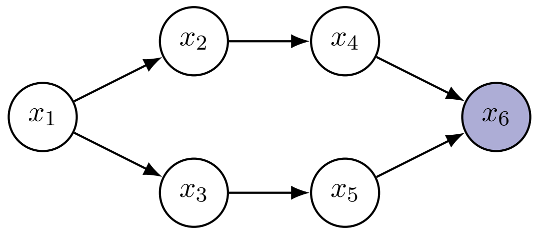
Some useful facts:

- $\log\det(K)$ is a strictly concave function of K.
- $\text{tr}(KS_n)$ is linear in K.
- The gradients are $\nabla_K \log\det(K) = K^{-1} = \Sigma$ and $\nabla_K \text{tr}(KS_n) = S_n$.
- The MLE is $\hat{\Sigma} = S_n$ (this is where the gradient vanishes).

4 Exercise 4: Variable Elimination

1. Simple Variable Elimination Example

Consider the following Directed Acyclic Graph (DAG) where we observe the variable $X_6 = \bar{x}_6$. We wish to compute the posterior $p(x_1|\bar{x}_6)$.



Factorization: The corresponding DAG model implies the factorization:

$$p(x_1, \ldots, x_6) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2)p(x_5|x_3)p(x_6|x_4, x_5)$$

Query: We want to compute $p(x_1|\bar{x}_6)$. We start by computing the marginal joint $p(x_1, \bar{x}_6)$ by eliminating the hidden variables $\mathbf{x}_R = \{x_2, x_3, x_4, x_5\}$.

$$p(x_1, \bar{x}_6) = \sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_5} p(x_1, \ldots, x_5, \bar{x}_6)$$

Using the **Variable Elimination** algorithm with the ordering $5, 4, 3, 2$:

$$\begin{aligned}
p(x_1, \bar{x}_6) &= p(x_1)\sum_{x_2} p(x_2|x_1)\sum_{x_3} p(x_3|x_1)\sum_{x_4} p(x_4|x_2)\underbrace{\sum_{x_5} p(x_5|x_3)p(\bar{x}_6|x_4, x_5)}_{\tau_1(x_3,x_4,\bar{x}_6)} \\
&= p(x_1)\sum_{x_2} p(x_2|x_1)\sum_{x_3} p(x_3|x_1)\underbrace{\sum_{x_4} p(x_4|x_2)\tau_1(x_3, x_4, \bar{x}_6)}_{\tau_2(x_2,x_3,\bar{x}_6)} \\
&= p(x_1)\sum_{x_2} p(x_2|x_1)\underbrace{\sum_{x_3} p(x_3|x_1)\tau_2(x_2, x_3, \bar{x}_6)}_{\tau_3(x_1,x_2,\bar{x}_6)} \\
&= p(x_1)\underbrace{\sum_{x_2} p(x_2|x_1)\tau_3(x_1, x_2, \bar{x}_6)}_{\tau_4(x_1,\bar{x}_6)}
\end{aligned}$$

Finally, we normalize:

$$p(x_1|\bar{x}_6) = \frac{p(x_1, \bar{x}_6)}{\sum_{x_1} p(x_1, \bar{x}_6)}$$

2. Complexity and Elimination Ordering

The computational complexity of Variable Elimination is $O(m \cdot k^{N_{\max}+1})$, where k is the number of states per variable and $N_{\max}$ is the maximum number of variables in a sum generated during the process. The ordering of variables crucially determines $N_{\max}$ and m is the number of factors.

Consider a model with the following factorization with $m = 8$:

$$p(C, D, \dots) \propto \phi(C)\phi(C, D)\phi(J, L, S)\phi(S, I)\phi(I)\phi(G, D, I)\phi(L, G)\phi(H, G, J)$$

Example 1: A "Bad" Ordering

Let's eliminate variables according to the ordering $\prec \{G, I, S, L, H, C, D\}$.

$$p(J) = \sum_D \sum_C \phi(C)\phi(C, D) \underbrace{\sum_H \sum_L \sum_S \phi(J, L, S) \underbrace{\sum_I \phi(S, I)\phi(I) \underbrace{\sum_G \phi(G, D, I)\phi(L, G)\phi(H, G, J)}_{\tau(D,L,H,J,I),\, N_G=6}}_{\tau(D,L,H,J,S),\, N_I=6}}_{\tau(D,J),\, N=5,4,\text{ then } 3}$$

(Simplification of the trace shown for brevity)

- The sum with the largest number of variables participating has $N_{\max} = 6$.
- **Complexity:** $O(8 \times k^6)$.

Example 2: A "Better" Ordering

Let's try the Elimination Ordering $\prec \{D, C, H, L, S, I, G\}$.

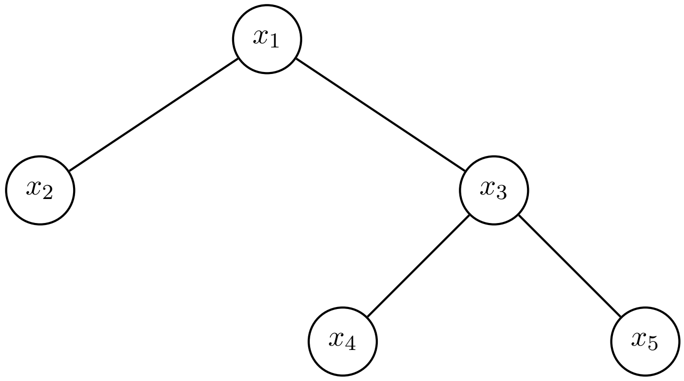
$$p(J) = \sum_G \sum_I \phi(I) \sum_S \phi(S, I) \sum_L \phi(L, G)\phi(J, L, S) \sum_H \phi(H, G, J) \sum_C \phi(C) \sum_D \phi(G, D, I)\phi(C, D)$$

Looking at the largest factor generated in this sequence, it is $\tau(G, I, J, L, S)$, $k_{\max} = 5$

- **Complexity:** $O(8 \times k^5)$.
- This demonstrates that choosing a good elimination ordering (finding the optimal one is NP-hard) significantly impacts inference speed.

5 Exercise 5: Sum-Product on Trees (Numerical Example)

Consider the following tree structure.



To have concrete numbers, suppose all variables are binary $x_i \in \{0,1\}$ and take unary potentials $\psi_i(x_i) = 1$. Let the pairwise potentials be defined by the following matrices (where rows/columns correspond to values 0 and 1):

$$\psi_{12} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, \quad \psi_{13} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \psi_{34} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}, \quad \psi_{35} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}.$$

In this notation, the (i,j)–entry of the matrix correspond to $\psi_{lk}(i,j)$.

1. Joint Distribution

The joint distribution is given by:

$$p(x_1, x_2, x_3, x_4, x_5) = \frac{1}{Z} \prod_{i=1}^{5} \psi_i(x_i) \psi_{12}(x_1, x_2) \psi_{13}(x_1, x_3) \psi_{34}(x_3, x_4) \psi_{35}(x_3, x_5).$$

Conditioning: Let's fix the values of three variables: $\bar{x}_2 = 1$, $\bar{x}_4 = 1$, $\bar{x}_5 = 0$. We get:

$$p(x_1, 1, x_3, 1, 0) = \frac{1}{Z} \psi_{12}(x_1, 1) \psi_{13}(x_1, x_3) \psi_{34}(x_3, 1) \psi_{35}(x_3, 0).$$

Direct Calculation: We compute the unnormalized probability values for the remaining free variables (x_1, x_3):

$$\begin{aligned}
p(0,1,0,1,0) &= \frac{1}{Z} \cdot 2 \cdot 2 \cdot 1 \cdot 1 = \frac{4}{Z} \\
p(0,1,1,1,0) &= \frac{1}{Z} \cdot 2 \cdot 1 \cdot 2 \cdot 1 = \frac{4}{Z} \\
p(1,1,0,1,0) &= \frac{1}{Z} \cdot 1 \cdot 1 \cdot 1 \cdot 1 = \frac{1}{Z} \\
p(1,1,1,1,0) &= \frac{1}{Z} \cdot 1 \cdot 2 \cdot 2 \cdot 1 = \frac{4}{Z}
\end{aligned}$$

Summing these terms (excluding Z): $4 + 4 + 1 + 4 = 13$. From this, we get the conditional distribution $p(x_1, x_3 | \bar{x}_2 = 1, \bar{x}_4 = 1, \bar{x}_5 = 0)$:

$$p(x_1, x_3 | \dots) = \frac{1}{13} \begin{bmatrix} 4 & 4 \\ 1 & 4 \end{bmatrix} \quad \text{(rows } x_1\text{, cols } x_3\text{)}$$

2. Message Passing (Marginal Distributions)

Suppose we are interested in the marginal distributions of x_1 and x_3. We compute the message passing formulas.

Messages from observed leaves:

$$m_{2\to1}(x_1) = \psi_2(1)\psi_{12}(x_1, 1) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \text{(Col 1 of } \psi_{12}\text{)}$$

$$m_{4\to3}(x_3) = \psi_4(1)\psi_{34}(x_3, 1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \text{(Col 1 of } \psi_{34}\text{)}$$

$$m_{5\to3}(x_3) = \psi_5(0)\psi_{35}(x_3, 0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{(Col 0 of } \psi_{35}\text{)}$$

Computing $m_{3\to1}(x_1)$: Since x_3 is not observed, we sum over it:

$$m_{3\to1}(x_1) = \sum_{x_3} \psi_3(x_3)\psi_{13}(x_1, x_3)m_{4\to3}(x_3)m_{5\to3}(x_3)$$

Calculating the product of incoming messages to node 3: $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$. Multiplying by transition ψ_{13}

$$m_{3\to1} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}. \quad \text{(matrix multiplication here)}$$

Belief for x_1:

$$b(x_1) = p(x_1|\bar{x}_2, \bar{x}_4, \bar{x}_5) \propto \psi_1(x_1)m_{2\to1}(x_1)m_{3\to1}(x_1)$$

$$b(x_1) \propto \begin{bmatrix} 2 \\ 1 \end{bmatrix} \odot \begin{bmatrix} 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 8 \\ 5 \end{bmatrix}.$$

Normalizing: $p(x_1 = 1| \dots) = \frac{5}{8+5} = \frac{5}{13}$.

Belief for x_3: To compute $b(x_3)$, we need the message from the other direction, $m_{1\to3}(x_3)$.

$$m_{1\to3}(x_3) = \sum_{x_1} \psi_1(x_1)\psi_{13}(x_1, x_3)m_{2\to1}(x_1)$$

Incoming to 1 is just $m_{2\to1} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. Multiplying by transition ψ_{13} (summing over x_1 means vector-matrix multiplication from left, or using symmetry):

$$m_{1\to3} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^T \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}.$$

Now, combine all messages arriving at node 3:

$$b(x_3) \propto \psi_3(x_3)m_{1\to3}(x_3)m_{4\to3}(x_3)m_{5\to3}(x_3)$$

$$b(x_3) \propto \begin{bmatrix} 5 \\ 4 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 2 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \end{bmatrix}.$$

Normalizing: $p(x_3 = 1|\bar{x}_2 = 1, \bar{x}_4 = 1, \bar{x}_5 = 0) = \frac{8}{5+8} = \frac{8}{13}$.