6.4110 PGM PSS Feb 13th

Probability Review

Belief (B): A probability distribution over state space $\mathcal{S}$.

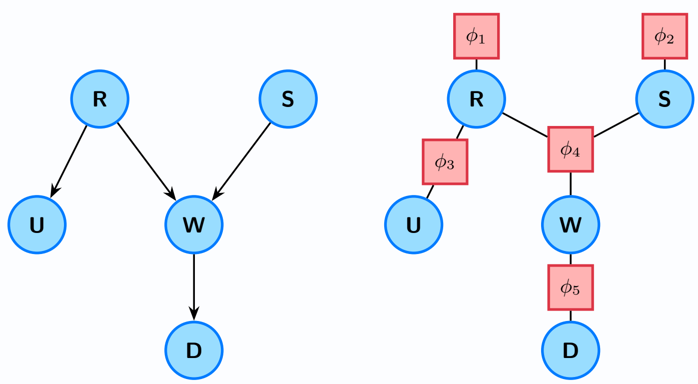
- **Bayes Rule:** Updates belief given observation E:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- **Distributions:**
  - **Joint:** $P(X, Y)$ (The full table of the probability for all possibilities).
  - **Marginal:** $P(X) = \sum_y P(X, y)$ (Summing out variables).
  - **Conditional:** $P(X|Y) = \frac{P(X,Y)}{P(Y)}$.

Bayesian Network vs. Factor Graph

Variables: Raining (R), Sprinklers On (S), Ground Wet (W), Dirt Tracks (D), Umbrella (U).



Bayes Net (Directed) **Factor Graph (Undirected)**

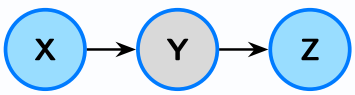
Bayes Nets	Factor Graphs
Nodes: Random Variables.	**Nodes:** Variables (Circles) & Factors (Squares).
Edges: Directed (Parent → Child).	**Edges:** Undirected (connect variable to factor).

What are factors? You can think of factors as tables that assign a compatibility score to every possible combination of values of the variables they involve. The higher this score, the more likely that particular set of values is to appear together. Factors can be any non-negative number (i.e. 67.67 is a valid factor). When we convert a Bayesian Network into a factor graph, we interpret the factors as conditional probabilities. This follows from the causality encoded the networks, and means we don't need to normalize. But it doesn't change the math since. $P(A|B) = Z_{A|B}P(A, B)$ where $Z_{A|B}$ is just a very specific constant, and we normalize into a probability distribution at the end anyway.

Paths of Dependence

Bayesian Networks: A path is *active* (dependent) or *blocked* (independent) based on three fundamental structures. **Conditioning on a node (gray)** changes the flow of information.

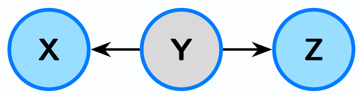
1. Chain (Cascade)



Unconditioned:
Path Active
(Dependent)

Condition on Y:
Path Blocked
(Independent)

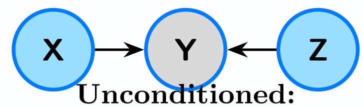
2. Fork (Common Cause)



Unconditioned:
Path Active
(Dependent)

Condition on Y:
Path Blocked
(Independent)

3. Collider (v-structure)

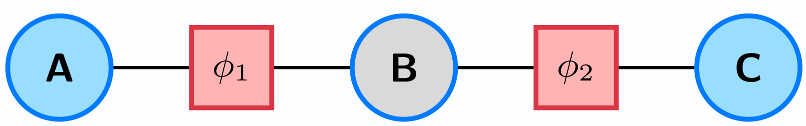


Unconditioned:
Path Blocked
(Independent)

Condition on Y (or child):
Path Active
(Dependent)

Factor Graphs (Separation): Rules are much simpler. Variables A and C are independent given B if *all paths* between A and C pass through an observed variable in B. Conditioning **always** blocks paths.

Observed Node Blocks Path

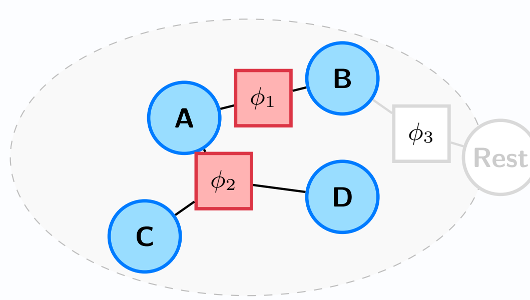


Conditioning separates A and C ($A \perp\!\!\!\perp C \mid B$)

Markov Blanket (MB): The minimal set of nodes needed to make a variable V independent of the rest of the graph.

- *Bayes Net:* Parents, Children, and Children's Parents. **Why Children's Parents?** Because conditioning on a child creates a v-structure (active path) between its parents. To block this active path, you must also condition on the other parent.
- *Factor Graph:* Immediate neighbors (variables connected to the exact same factors that V touches).

Example Factor Graph Markov Blanket



Belief Propagation (Sum-Product Algorithm)

Goal: Efficiently compute marginals $P(V_i)$ on tree-structured graphs.
Mechanism: Passing messages (μ) between nodes from leaves to root, then from root to leaves. Start with the messages from leaves equal to 1 if they have no unitary factors (otherwise set them equal to the factor). Note on notation: the $\sum$ and $\prod$ represent variable elimination and factor multiplication, respectively. This **is not** matrix multiplication, or any element-wise operation.

1. **Factor to Variable ($\mu_{\phi \to V}$):** The factor summarizes its information for variable V by summing out all other variables connected to it.

$$\mu_{\phi \to V}(v) = \sum_{\mathbf{x} \setminus v} \left(\phi(v, \mathbf{x}) \prod_{y \in N(\phi) \setminus V} \mu_{y \to \phi} \right)$$

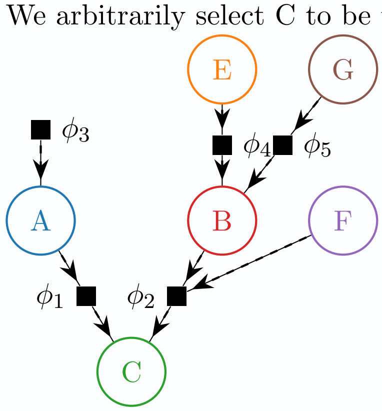
2. **Variable to Factor ($\mu_{V \to \phi}$):** The message aggregates all incoming information to the variable, excluding the target factor, then passes it to ϕ. This directionality is why the algorithm only works on trees.

$$\mu_{V \to \phi}(v) = \prod_{\psi \in N(V) \setminus \phi} \mu_{\psi \to V}(v)$$

Final Belief: $P(V) \propto \prod_{\phi \in N(V)} \mu_{\phi \to V}(v)$.
Handling evidence? Add a factor to the variable you have knowledge of with $P(E = e) = 1$.
Not a tree? This algorithm only works on trees. You need to do a trick to turn it into a tree.

Forward Pass (Leaves to Root)

We arbitrarily select C to be the root and propagate information to it. We want to find P(B).



$$\mu_{\phi_3 \to \mathbf{A}} = \phi_3, \quad \mu_{\mathbf{A} \to \phi_1} = \mu_{\phi_3 \to \mathbf{A}}$$

$$\mu_{\phi_1 \to \mathbf{C}} = \sum_{\mathbf{A}} \phi_1 \cdot \mu_{\mathbf{A} \to \phi_1}$$

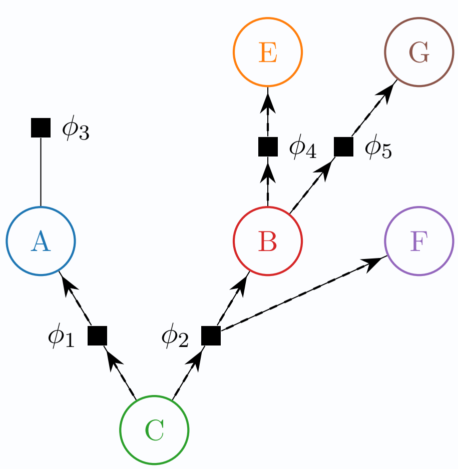
$$\mu_{\phi_4 \to \mathbf{B}} = \sum_{\mathbf{E}} \phi_4 \cdot \mathbf{1}, \quad \mu_{\phi_5 \to \mathbf{B}} = \sum_{\mathbf{G}} \phi_5 \cdot \mathbf{1}$$

$$\mu_{\mathbf{B} \to \phi_2} = \mu_{\phi_4 \to \mathbf{B}} \cdot \mu_{\phi_5 \to \mathbf{B}}$$

$$\mu_{\phi_2 \to \mathbf{C}} = \sum_{\mathbf{B}, \mathbf{F}} \phi_2 \cdot \mu_{\mathbf{B} \to \phi_2} \cdot \mathbf{1}$$

$$P(\mathbf{C}) \propto \mu_{\phi_1 \to \mathbf{C}} \cdot \mu_{\phi_2 \to \mathbf{C}}$$

Backward Pass (Root to Leaves)



$$\mu_{\mathbf{C} \to \phi_2} = \mu_{\phi_1 \to \mathbf{C}}, \quad \mu_{\mathbf{C} \to \phi_1} = \mu_{\phi_2 \to \mathbf{C}}$$

$$\mu_{\phi_2 \to \mathbf{B}} = \sum_{\mathbf{C}, \mathbf{F}} \phi_2 \cdot \mu_{\mathbf{C} \to \phi_2} \cdot \mathbf{1}$$

$$\mu_{\mathbf{B} \to \phi_4} = \mu_{\phi_2 \to \mathbf{B}} \cdot \mu_{\phi_5 \to \mathbf{B}}$$

$$\mu_{\phi_4 \to \mathbf{E}} = \sum_{\mathbf{B}} \phi_4 \cdot \mu_{\mathbf{B} \to \phi_4}$$

$$\mu_{\phi_1 \to \mathbf{A}} = \sum_{\mathbf{C}} \phi_1 \cdot \mu_{\mathbf{C} \to \phi_1}$$

$$P(\mathbf{B}) \propto \mu_{\phi_4 \to \mathbf{B}} \cdot \mu_{\phi_5 \to \mathbf{B}} \cdot \mu_{\phi_2 \to \mathbf{B}}$$

Factor Multiplication

Definition: Factor multiplication (or product) combines two factors ϕ_1 and ϕ_2 into a new factor ψ. This is the fundamental operation for "joining" information from different parts of a graph.

$$\psi(A, B, C) = \phi_1(A, B) \times \phi_2(B, C)$$

The Procedure:

1. **Identify Scope:** The new factor's scope is the union of the inputs: $\{A, B\} \cup \{B, C\} \to \{A, B, C\}$.
2. **Match Shared Variables:** Iterate through every possible combination of A, B, C.
3. **Multiply Values:** For a specific row (a, b, c), find the entry in ϕ_1 matching (a, b) and the entry in ϕ_2 matching (b, c), then multiply them.

Example: Let A, B, C be binary $\{0, 1\}$.

Factor $\phi_1(A, B)$

A	B	Val
0	0	0.3
0	1	0.7
1	0	0.2
1	1	0.8

$\times$

Factor $\phi_2(B, C)$

B	C	Val
0	0	0.1
0	1	0.9
1	0	0.4
1	1	0.6

Resulting Factor $\psi(A, B, C)$:

Variable Elimination

Goal: Compute $P(C)$ from a factor graph with factors $\phi_1(A, B)$ and $\phi_2(B, C)$.
Joint (unnormalized):

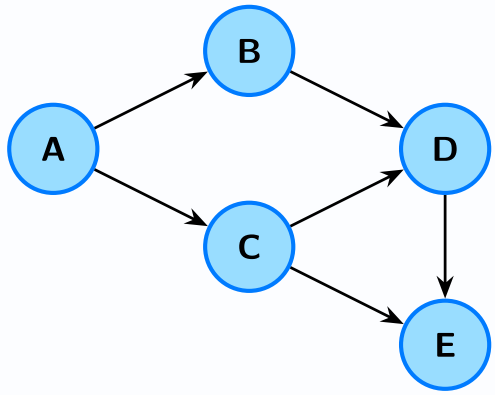
$$P(A, B, C) \propto \phi_1(A, B) \ \times \phi_2(B, C).$$

A	B	$\phi_1(A, B)$	B	C	$\phi_2(B, C)$
0	0	0.3	0	0	0.1
0	1	0.7	0	1	0.9
1	0	0.2	1	0	0.4
1	1	0.8	1	1	0.6

$$P(C) = \sum_{A,B} \phi_1(A, B) \times \phi_1(B, C)/Z.$$

Path Dependence Practice

Example Bayesian Network:



Question: Is $A \perp E \mid C, B$?

Question: Is $C \perp B \mid A$?

Question: Is $C \perp B \mid D$?

Question: Is $C \perp D \mid E$?

Question: Is $B \perp C$?

Question: Is $B \perp E$?

Question: Is $B \perp E \mid A, D$?

Belief Propagation Practice

Question: Convert the Bayesian Network in the prior part to a Factor Graph.

Question: Can you run belief propagation on this? Why or why not?

Question: Assign the factors the following values (note this is no longer a valid Bayesian Network) and perform Belief Propagation to find the probability that B is true given that C is True.

A	$\phi_1(A)$
0	3
1	1

A	B	$\phi_2(A, B)$
0	0	2
0	1	1
1	0	1
1	1	2

A	C	$\phi_3(A, C)$
0	0	3
0	1	1
1	0	1
1	1	3

B	C	D	$\phi_4(B, C, D)$
0	0	0	2
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	1
1	0	1	2
1	1	0	1
1	1	1	1

C	D	E	$\phi_5(C, D, E)$
0	0	0	2
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	1
1	0	1	1
1	1	0	4
1	1	1	1