Reasoning (I)

Mingsheng Long

Tsinghua University

Spring 2023

Outline

- **Probabilistic Reasoning**
- Graphical Models
  - Bayesian Networks
  - Markov Random Fields
- Conditional Independence
  - D-separation
  - Markov Properties
- Inference in Graphical Models
  - Variable Elimination
  - Belief Propagation

Modeling and Inference

- How does a weather forecast predict the probability of Sunshine (S) or Rain (R) given the observation of Temperature (T) and Date (D)?
- **Modeling:**

  Find the **joint** distribution
  $$p(S, R, T, D)$$

- **Inference:**
  - **Condition** on evidence (Temperature and Date):
  $$T = 36.5℃, D = 2021/7/31$$
  - Interested in certain variable set (**query**), such as rain ($\{R\}$):

  $$p(\, R \mid T = 36.5℃, D = 2021/7/31)$$
  (S is marginalized out)

Probabilistic Reasoning

- **Probabilistic Reasoning = Modeling + Inference**
  - $X = \{x_1, \dots, x_D\}$ is a set of D random variables.
  - Query set R and condition set C are subsets of X.
- **Modeling**: How to specify a joint distribution $p(x_1, \dots, x_D)$ **compactly**?
  - Bayesian networks
  - Markov random fields
- **Inference**: How to compute $p(R \mid C)$ **efficiently**?
  - Elimination methods (变量消除法)——lec9
  - Latent variable models (隐变量模型)——lec10
  - Variational methods (变分方法)——lec11
  - Sampling methods (采样方法)——lec12

Outline

- Probabilistic Reasoning
- **Graphical Models**
  - **Bayesian Networks**
  - Markov Random Fields
- Conditional Independence
  - D-separation
  - Markov Properties
- Inference in Graphical Models
  - Variable Elimination
  - Belief Propagation

Bayesian Network

- Product rule of probability implies joint distribution of K variables:

$$p(x_1, \dots, x_K) = p(x_K | x_1, \dots, x_{K-1}) \cdots p(x_2 | x_1) p(x_1).$$

4 nodes example

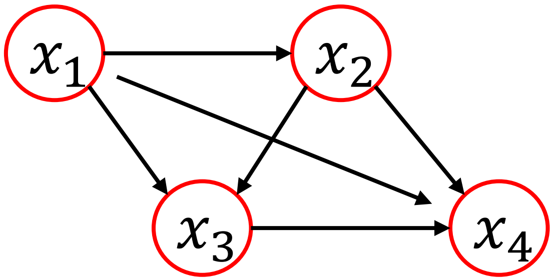
$x_1 | \emptyset$

$x_2 | x_1$

$x_3 | x_1, x_2$

$x_4 | x_1, x_2, x_3$

Induce directed graph relations



- A **Bayesian network** is a **directed acyclic graph (DAG)** that specifies a joint distribution as a product of local conditional distributions, one for each node:

$$p(x_1, \dots, x_K) = \prod_{s=1}^{K} p\left(x_s \middle| \boldsymbol{x}_{\Gamma(s)}\right)$$

where $\Gamma(s)$ denotes the set of parents of x_s.

Example: The Alarm Network

Q: Earthquakes (E) and burglaries (B) are independent events that will cause an alarm (A) to go off. Suppose you hear an alarm. How does hearing the earthquake report (R) change your beliefs about burglary?

Example: The Alarm Network

- Choosing an **ordering**:

$$p(A,R,E,B) = p(A|R,E,B)p(R|E,B)p(E|B)p(B)$$

- **Assumptions:**
  - The alarm is not influenced by the radio report.

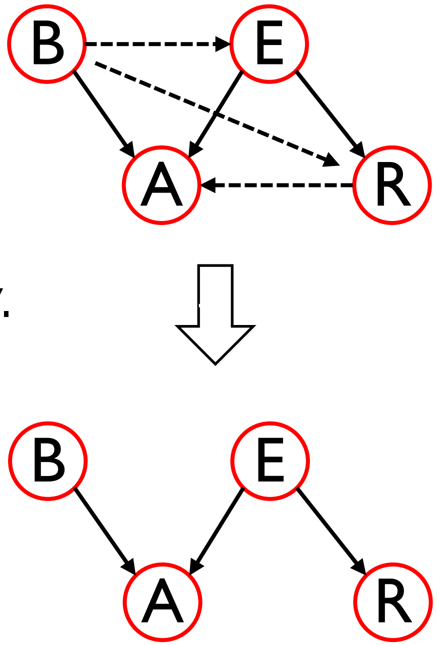
$$p(A|R,E,B) = p(A|E,B)$$

  - The radio does not broadcast any burglary activity.

$$p(R|E,B) = p(R|E)$$

  - Burglaries don't cause earthquakes.

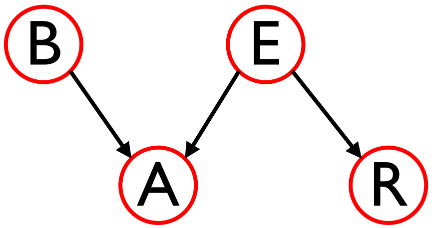
$$p(E|B) = p(E)$$



- **Modeling:** $p(A,R,E,B) = p(A|E,B)p(R|E)p(E)p(B)$

Example: The Alarm Network

- **Parameters:**
  - $p(B=1) = p(E=1) = 0.01$
  - $B \vee E \Longrightarrow (A=1)$ almost surely
  - $E \Longrightarrow R$ almost surely



- **Inference:**
  - Probability of Burglary only given Alarm:

$$p(B=1|A=1) = \frac{\sum_{E,R} p(B=1,E,A=1,R)}{\sum_{B,E,R} p(B,E,A=1,R)} \approx 0.5$$

Earthquakes or Burglaries almost half and half

Explaining away

  - Probability of Burglary given Alarm and Radio:

$$p(B=1|A=1,R=1) = 0.01$$

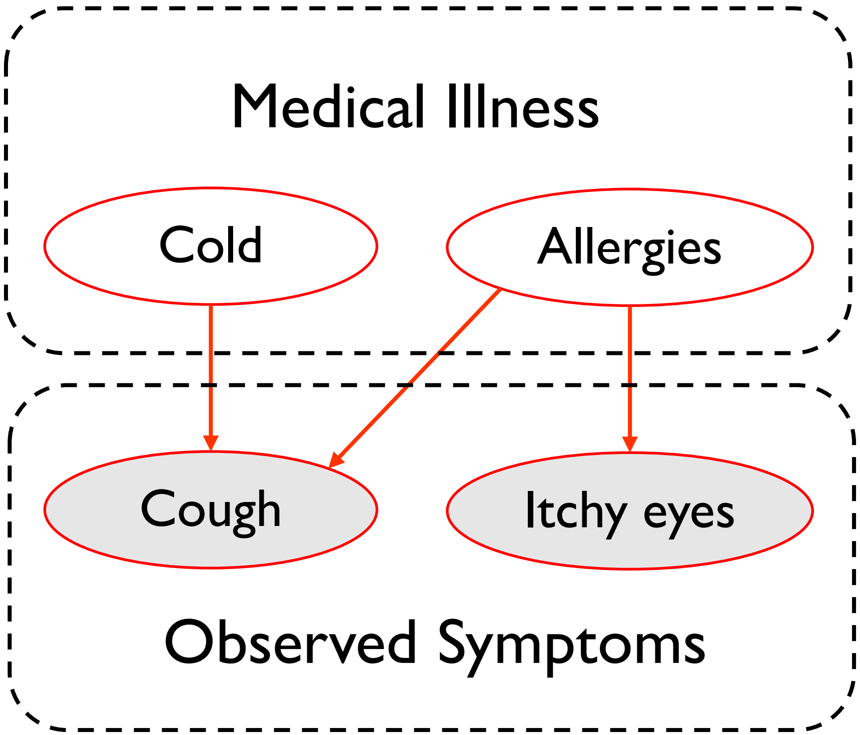
Nearly sure the Earthquakes

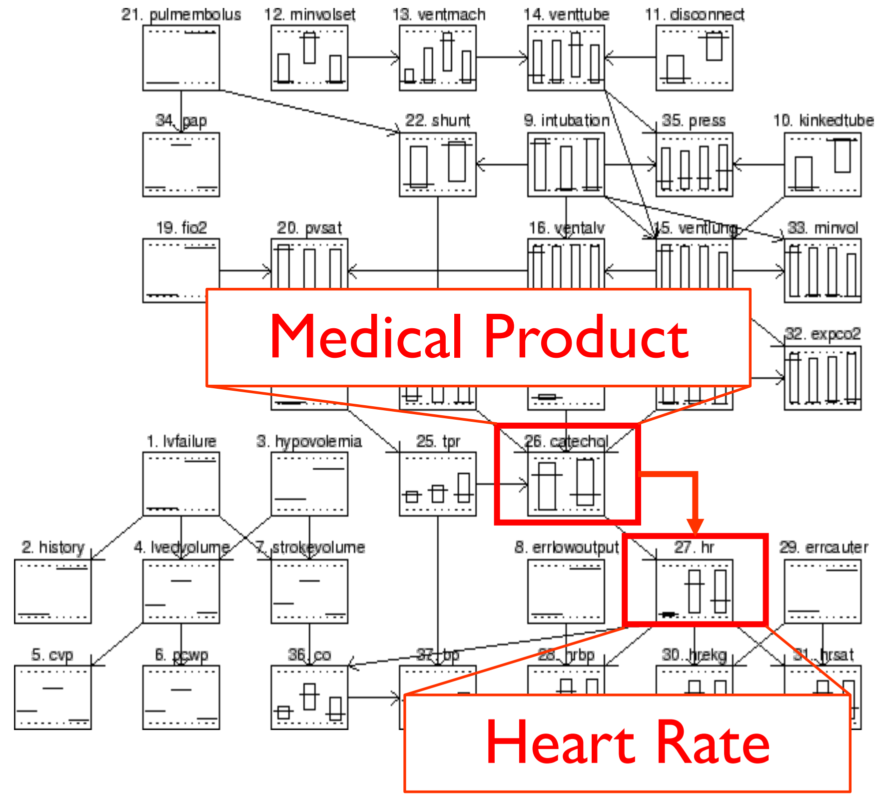
Example: Medical Diagnosis

Explainable

Medical illness diagnoses

ICU monitoring





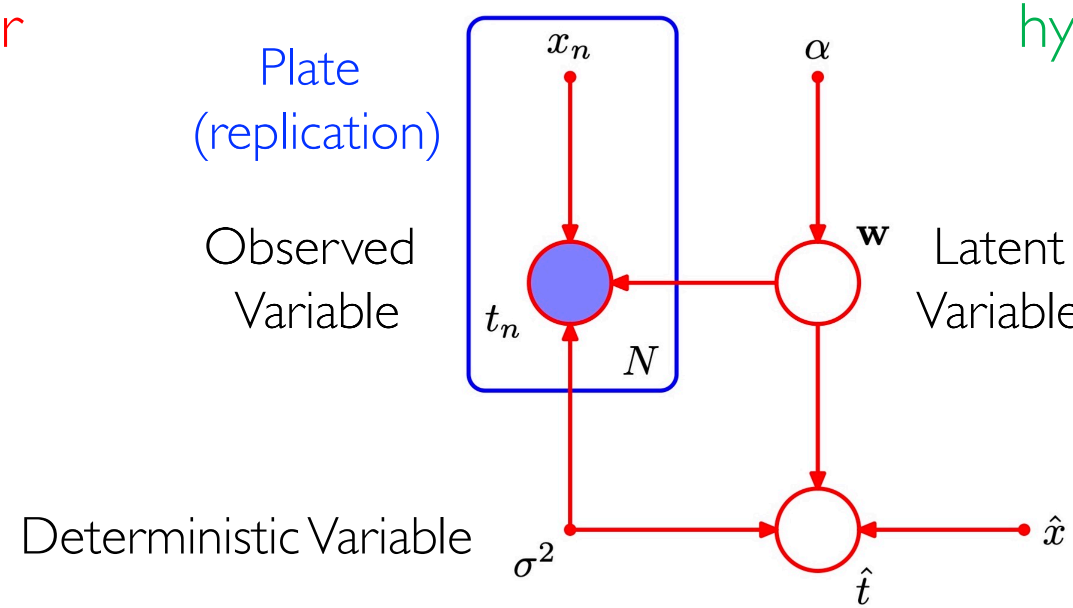
Example: Polynomial Regression

- The polynomial regression model, jointly showing also a new input value $\hat{x}$ together with the corresponding model prediction $\hat{t}$.

$$p(\hat{t}, \boldsymbol{t}_{1:N}, \boldsymbol{w} | \hat{x}, \boldsymbol{x}_{1:N}, \alpha, \sigma^2) = \left[\prod_{n=1}^{N} p(t_n | x_n, \boldsymbol{w}, \sigma^2)\right] \times p(\boldsymbol{w}|\alpha) p(\hat{t}|\hat{x}, \boldsymbol{w}, \sigma^2)$$

parameter

hyperparameter



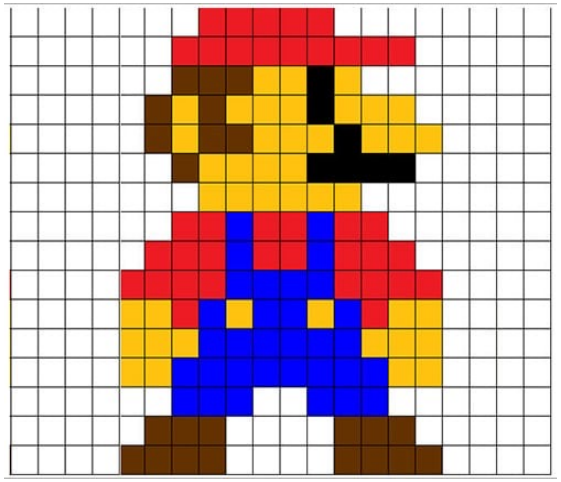
Outline

- Probabilistic Reasoning
- **Graphical Models**
  - Bayesian Networks
  - **Markov Random Fields**
- Conditional Independence
  - D-separation
  - Markov Properties
- Inference in Graphical Models
  - Variable Elimination
  - Belief Propagation

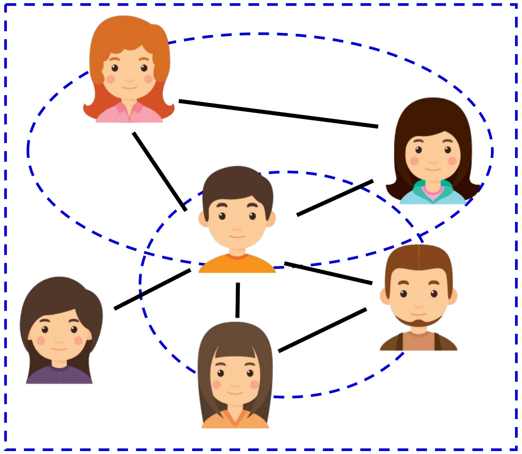
Are Bayesian Network Enough

- Bayesian networks (Directed acyclic graph, DAG) can hardly model the **bi-directional or cycling relationships**.

Adjacency between Pixels in a Figure



Friend relation in Social Networks



Example: Voting Preferences

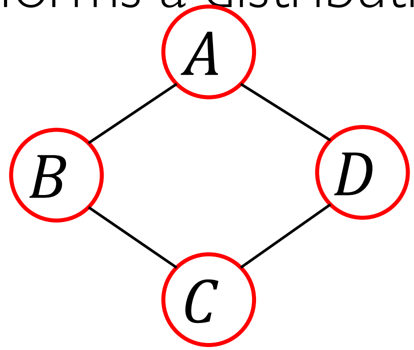
- Four students A, B, C, D are voting for a group leader.
- Say that $(A, B), (B, C), (C, D)$, and (D, A) are friends, and friends tend to have similar voting preference.
- Friend relationships can be naturally represented by an undirected graph.
  - **Identify dependent variables:**

$$p(A, B, C, D) = \frac{1}{Z} \phi(A, B) \phi(B, C) \phi(C, D) \phi(D, A)$$

where Z is a normalizing constant to ensure p forms a distribution.

  - **Define the strength of interactions:**

$$\phi(X, Y) = \begin{cases} 10 & \text{if } X = Y = 1 \\ 5 & \text{if } X = Y = 0 \\ 1 & \text{otherwise} \end{cases}$$

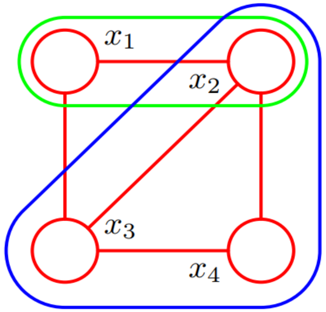


Markov Random Field

- A **Markov Random Field (MRF)** is an **undirected graph** that specifies a joint distribution as a product of potential functions:

$$p(\boldsymbol{x}) = \frac{1}{Z}\prod_{C}\psi_C(\boldsymbol{x}_C)$$

  - C denotes a **clique** (fully connected subgraphs) in graph G.
  - $\boldsymbol{x}$ denotes $\{x_1 \ldots\ldots x_n\}$, $\boldsymbol{x}_C$ denotes $\{x_i \mid x_i \in C\}$.
  - Each ψ_C is a non-negative function over the variables in a clique.
  - Z is the normalizing constant (partition function).
- We cannot factorize the potential function of a fully connected subgraph.



Energy Function

- Because we are restricted to strictly positive potential functions, it is convenient to express them as exponentials.

$$\psi_C(\boldsymbol{x}_C) = \exp\{-E(\boldsymbol{x}_C)\}$$

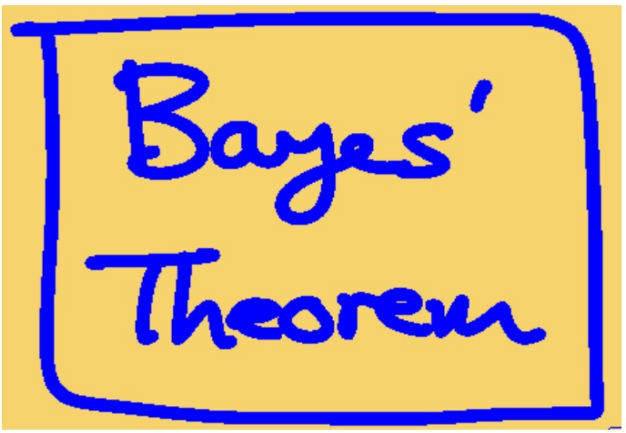
- $E(\boldsymbol{x}_C)$ is called an energy function, and so the total energy is obtained by adding the energies of each cliques.

$$p(\boldsymbol{x}) = \frac{1}{Z}\prod_C \psi_C(\boldsymbol{x}_C)$$
$$= \frac{1}{Z}\exp\{-\sum_C E(\boldsymbol{x}_C)\}$$

$\boxed{\text{Lower energy} \Rightarrow \text{Higher probability}}$

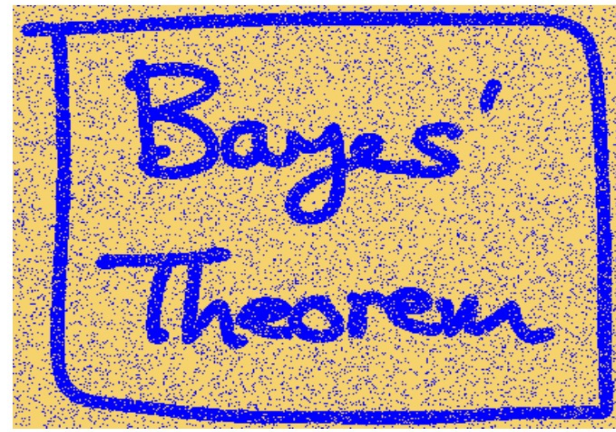
Example: Image Denoising

Denoising



Noise-free image

$x_i \in \{-1, +1\}$



Observed image

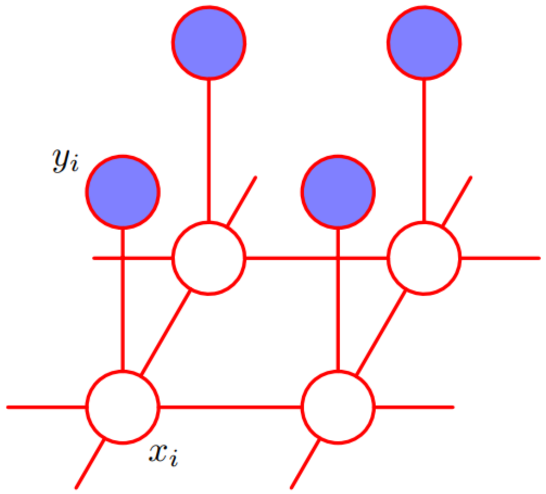
$y_i \in \{-1, +1\}$

Flipping the sign with probability 10%

Example: Image Denoising

- Modeling image with an **MRF**:
  - **Nodes:** pixels in both images x_i, y_i.
  - **Edges:** (x_i, x_j) for each neighbor pair i, j;

    (x_i, y_i) for each position i.



- Define the energy function (Pixels with same sigh ⇒ Lower energy)
  - Pixels at same position x_i and y_i have strong correlation.

$$E(x_i, y_i) = -\eta x_i y_i$$

  - Neighboring pixels x_i and x_j in an image are strongly correlated.

$$E(x_i, x_j) = -\beta x_i x_j$$

Example: Image Denoising

- The complete energy function

$$E(\boldsymbol{x}, \boldsymbol{y}) = -\eta \sum_{i} x_i y_i - \beta \sum_{\{i,j\}} x_i x_j$$

- The joint distribution

$$p(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{Z} \exp\{-E(\boldsymbol{x}, \boldsymbol{y})\}$$

- Pixels of observed image → $\boldsymbol{y}$.
- Denosing: find noise-free image $\boldsymbol{x}$ with high probability given the observation $\boldsymbol{y}$.

$$\widehat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} p(\, \boldsymbol{x} \mid \boldsymbol{y} \,)$$

Example: Image Denoising

- **Algorithm (Iterated conditional modes, ICM)**
  1. Initialize the variables $\{x_i\}$ (e.g. by $x_i = y_i$ for all i).
  2. For each pixel x_i :

     Evaluate the total energy of possible states $x_i = +1$ and $x_i = -1$ keeping all other node fixed.

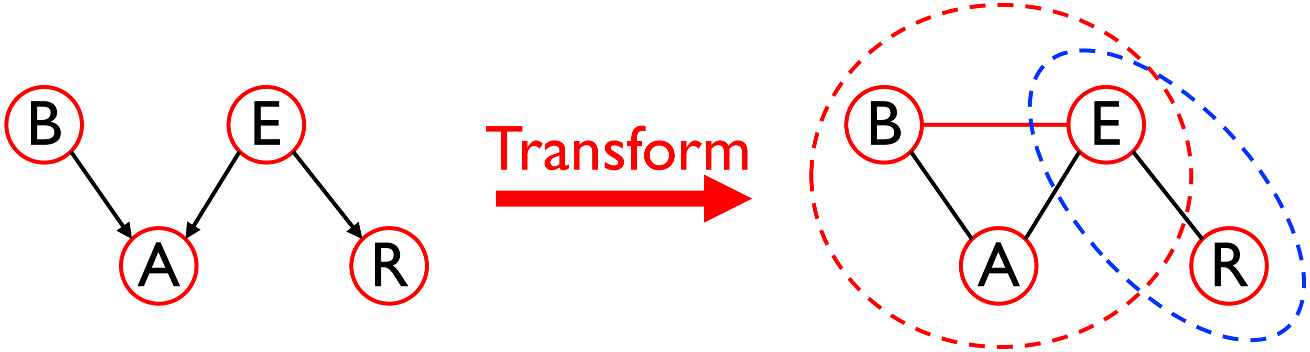
     Set x_i to whichever state has the lower energy.

$$x_i^{\text{new}} = \underset{x_i \in \{0,1\}}{\operatorname{argmax}}\, p\left(x_i \mid \boldsymbol{y}, \boldsymbol{x}^{\text{old}} \right)$$

  3. Repeat step 2. until converge.

Greedy Method!

Relation to Directed Graphs



$$p(B)p(E)\underbrace{p(A \mid B,E)}_{\text{Cannot decompensate}}p(R \mid E)$$

$$\phi_1 = p(B)p(E)p(A \mid B,E)$$

$$\phi_2 = p(R \mid E)$$

- Bayesian network can be transformed into MRF by
  - Linking the parents of each node;
  - Removing the direction of each arrow.
- The transformation method is not unique.

Outline

- Probabilistic Reasoning
- Graphical Models
  - Bayesian Networks
  - Markov Random Fields
- **Conditional Independence**
  - **D-separation**
  - Markov Properties
- Inference in Graphical Models
  - Variable Elimination
  - Belief Propagation

Conditional Independence

- X and Y are independent random variables if and only if

$$p_{XY}(x, y) = p_X(x) p_Y(y)$$

for all $x \in \mathcal{X}, y \in \mathcal{Y}$, denoted as $X \perp Y$.

- X and Y are conditionally independent random variables given another conditioning variable Z (also known as "observed variable")

$$p_{XY|Z}(x, y \mid z) = p_{X|Z}(x \mid z)\, p_{Y|Z}(y \mid z)$$

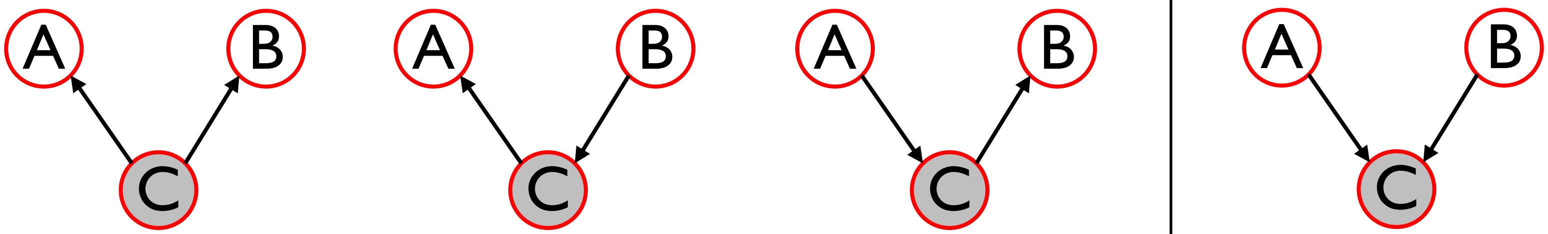
or alternatively,

$$p_{X|YZ}(x \mid y, z) = p_{X|Z}(x \mid z)$$

for all $x \in \mathcal{X}, y \in \mathcal{Y}$ and $z \in \mathcal{Z}$, denoted as $X \perp Y \mid Z$.

Marginal Independence in Directed Models

$(A \perp B)$



(a) tail-to-tail **(b) head-to-tail** **(c) head-to-tail** **(d) head-to-head**

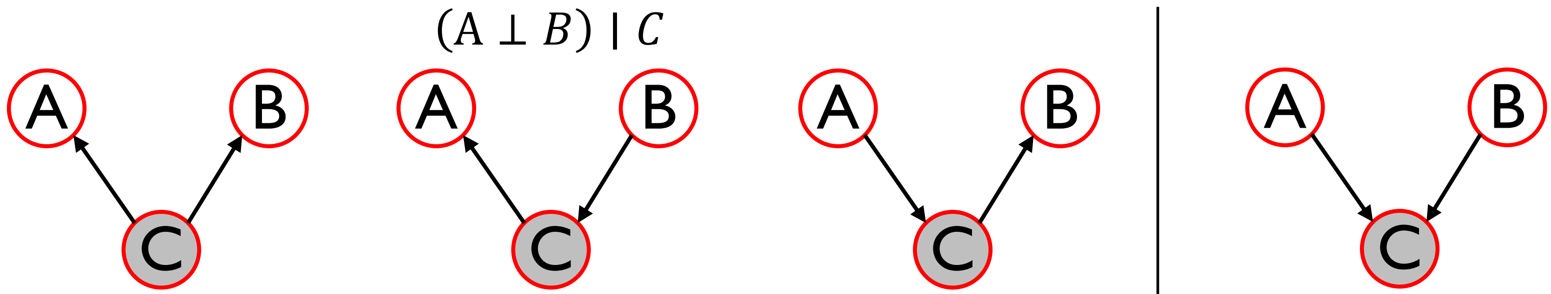
- In (a), (b) and (c), A and B are dependent random variables.
- In (d), A and B are independent random variables.

Definition of Bayesian networks

$$p(A,B) = \sum_C p(A,B,C) = \sum_C p(A)p(B)p(C|A,B)$$

$$= p(A)p(B)\sum_C p(C|A,B) = p(A)p(B)$$

Conditional Independence in Directed Models



(a) tail-to-tail (b) head-to-tail (c) head-to-tail (d) head-to-head

- In (a), (b) and (c), A and B are conditionally independent given C.
  - (a) $p(A, B \mid C) = \frac{p(A|C)\, p(B|C)\, p(C)}{p(C)} = p(A \mid C)\, p(B \mid C)$
  - (b) $p(A, B \mid C) = \frac{p(A|C)\, p(C|B)\, p(B)}{p(C)} = p(A \mid C)\, p(B \mid C)$

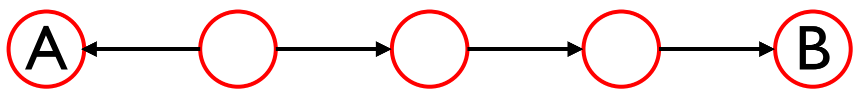
Bayes formula

- In general case of (d), A and B are conditionally dependent given C.
  - (d) $p(A, B \mid C) \propto p(C \mid A, B)\, p(A)\, p(B)$

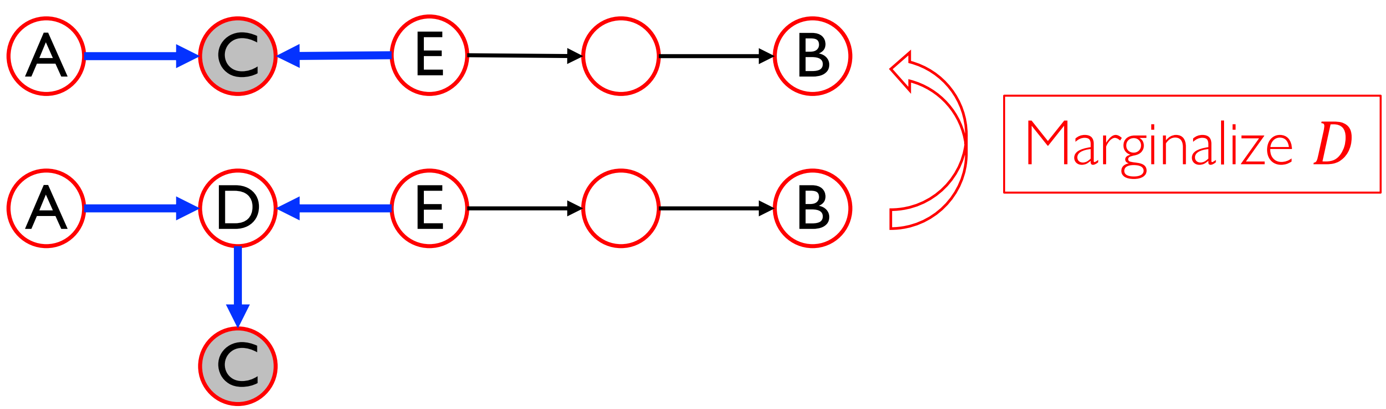
Marginally independent but conditionally dependent!

Conditional Dependence Criteria

- Consider an undirected path U between A and B, the path is **connected** given condition set C if it belongs to one of these cases:
  - Non-head-to-head nodes of U do not belong to C.



  - Any head-to-head node belongs to C or it has descendant that belongs to C.

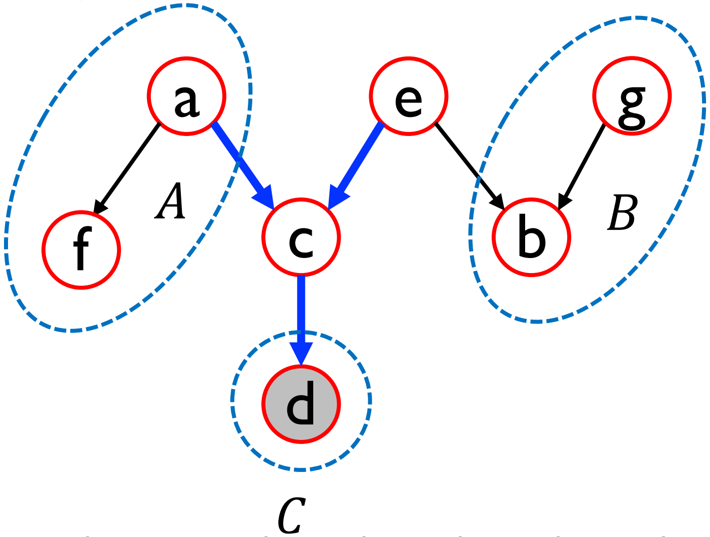


D-separation

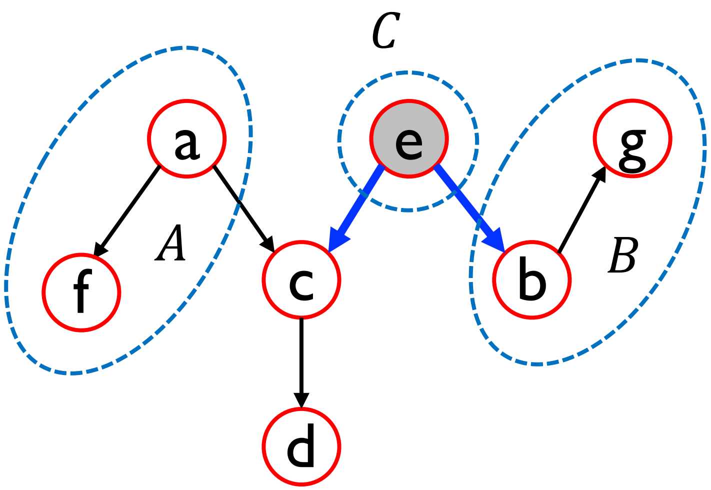
- All paths from any node in A (node set) to any node in B (node set) are **disconnected** given C.

  $\Leftrightarrow$ A is said to be D-separated from B by C.

  $\Leftrightarrow$ $(A \perp B) \mid C$



Although c is a head-to-head node, a and b are connected due to d is observed.



D-separated!

Markov Blanket

- How to infer a node with variable $\boldsymbol{x}_i$ from the remaining variables $\boldsymbol{x}_{j\neq i}$ in directed models ?

$p(\,x_i \mid \boldsymbol{x}_{\{j\neq i\}}\,)$ Worst case: Depend on all nodes of the graph

- Simplify: can we use part of the graph for inference?
- Find $S_1 \in \boldsymbol{x}_{\{j\neq i\}}$ which contents

$$x_i \perp (\boldsymbol{x}_{\{j\neq i\}} \setminus S_1) \mid S_1$$

- It means that S_1 contains all information one needs to infer x_i.

$$p(\,x_i \mid \boldsymbol{x}_{\{j\neq i\}}\,) = p(\,x_i \mid S_1 \sqcup (\boldsymbol{x}_{\{j\neq i\}} \setminus S_1)\,) = p(x_i \mid S_1)$$

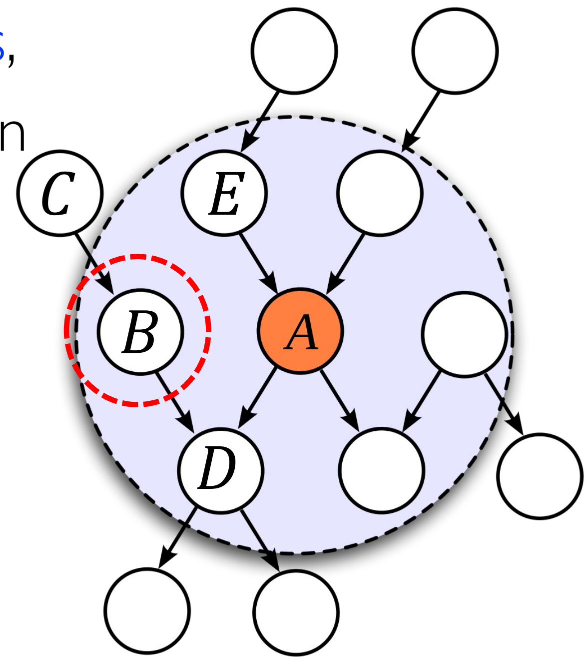
- S_1 is called the Markov blanket of x_i.

Markov Boundary

- How to infer a node with variable x_i from the remaining variables $x_{j\neq i}$ in directed models ?
- A Markov boundary is the minimal Markov blanket. In a Bayesian network, it includes parents, children and the other parents of all of its children (co-parents).

$$p(x_i \mid \boldsymbol{x}_{\{j\neq i\}}) = p(x_i \mid x_{\mathrm{pa}} \cup x_{\mathrm{ch}} \cup x_{\mathrm{co\cdot pa}})$$

- $B \not\perp A \mid D$ while $C \perp A \mid B, D$



Markov Boundary 'block' every path from A to outside nodes.

Outline

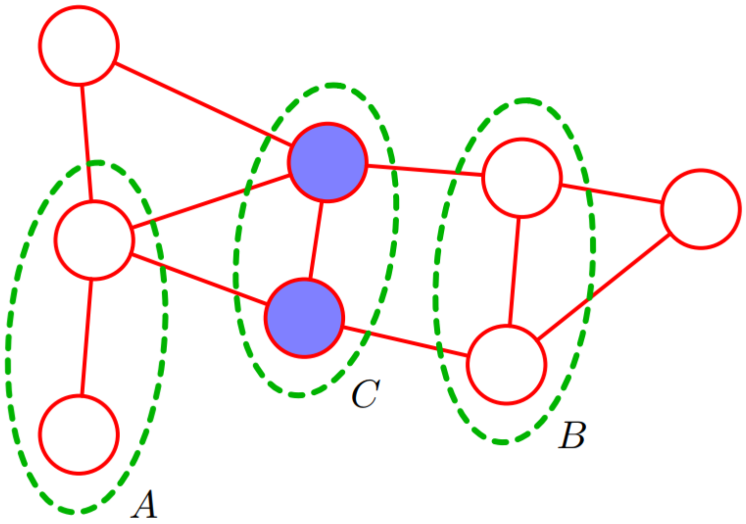
- Probabilistic Reasoning
- Graphical Models
  - Bayesian Networks
  - Markov Random Fields
- **Conditional Independence**
  - D-separation
  - **Markov Properties**
- Inference in Graphical Models
  - Variable Elimination
  - Belief Propagation

Markov Properties

- What independencies can be described by an undirected MRF?
  - x, y are dependent if they are connected by a path of unobserved variables.

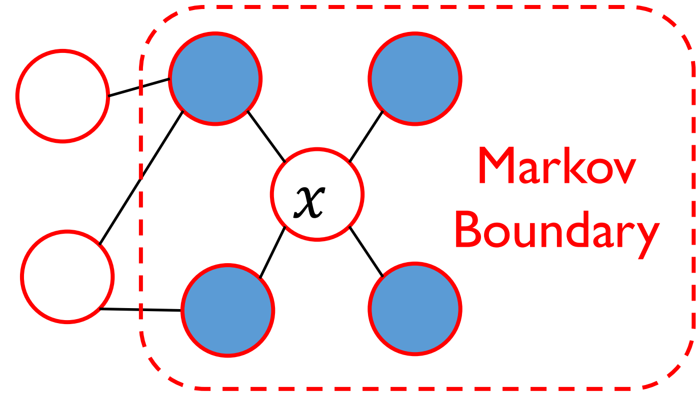
Global Markov Property:
If any path from $\boldsymbol{A}$ to $\boldsymbol{B}$ passes through $\boldsymbol{C}$, which denoted as $\mathrm{sep}_G(\boldsymbol{A}, \boldsymbol{B} \mid \boldsymbol{C})$, then

$$(\boldsymbol{X}_A \perp \boldsymbol{X}_B) \mid \boldsymbol{X}_C$$



Markov Properties

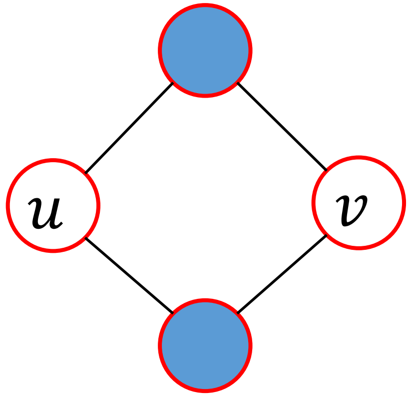
- If x's neighbors are all observed, then x is independent of all the other variables.



Local Markov Property:
For any node x, $N(x)$ denotes neighbors of x, $N[x] = x \cup N(x)$, then

$$X_x \perp \boldsymbol{X}_{V \setminus \mathrm{N}[x]} \mid \boldsymbol{X}_{N(x)}$$

- Suppose u and v are not linked, they are independent when all the other variables are observed.



Pairwise Markov Property:
If u and v are not linked, then

$$X_u \perp X_v \mid \boldsymbol{X}_{V \setminus \{u,v\}}$$

Markov Random Fields

- $\boldsymbol{X}$ forms a Markov random fields with respect to graph $\boldsymbol{G} = (V, E)$ if they satisfy the Markov properties:

  (a) Global Markov property:

  $$\mathrm{sep}_G(\boldsymbol{A}; \boldsymbol{B} \mid \boldsymbol{C}) \Rightarrow (\boldsymbol{X}_{\boldsymbol{A}} \perp \boldsymbol{X}_{\boldsymbol{B}}) \mid \boldsymbol{X}_{\boldsymbol{C}} \text{ for all disjoint set } \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$$

  (b) Local Markov property:

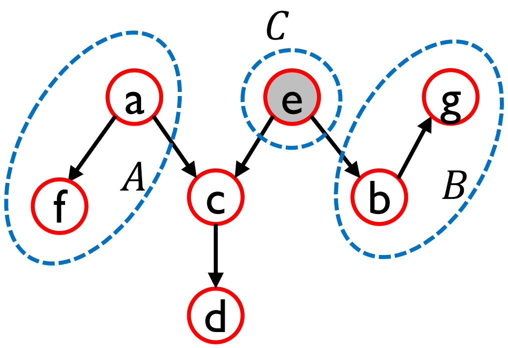
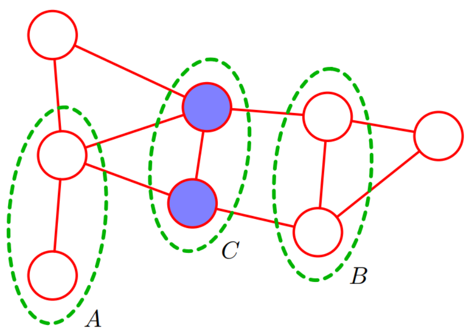
  $$X_v \perp \boldsymbol{X}_{V \setminus \mathrm{N}[v]} \mid \boldsymbol{X}_{N(v)} \text{ for } \forall v \in V$$

  (c) Pairwise Markov property:

  $$X_i \perp X_j \mid \boldsymbol{X}_{V \setminus \{i,j\}} \text{ for } \forall (i, j) \notin E$$

- **Theorem:** the above three Markov properties are equivalent for a positive probability distribution.

Summary for Graphical Modeling

	Factorization Properties	Independent Properties
Bayesian Network	$p(\boldsymbol{x}) = \prod_{s=1}^{N} p(x_s \mid \boldsymbol{x}_{\Gamma(s)})$ where $\Gamma(s)$ denotes the set of parents of x_s.	D-separation 
Markov Random Field	$p(x) = \frac{1}{Z} \prod_{C} \phi_C(\boldsymbol{x}_C)$ $\phi_C(\boldsymbol{x}_C)$ are potential functions of cliques.	Markov Properties 

Outline

- Probabilistic Reasoning
- Graphical Models
  - Bayesian Networks
  - Markov Random Fields
- Conditional Independence
  - D-separation
  - Markov Properties
- **Inference in Graphical Models**
  - **Variable Elimination**
  - Belief Propagation

Probabilistic Inference

- Marginal inference
  - What is the probability of a given variable in our model after we sum everything else out?

$$p(y=1) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} p(y=1, x_1, x_2, \dots, x_n)$$

- Maximum a posteriori (MAP) inference
  - What is the most likely assignment to the variables in the model?

$$\max_{x_1,\dots,x_n} p(y=1, x_1, \dots, x_n)$$

- It is **NP-hard** to answer these questions exactly for general cases.

Example: Markov Chain

- **Discrete assumption:** $x_1, \ldots, x_n$ are discrete taking k possible values.
- Given a chain Bayesian network (Markov chain).

$$x_1 \to x_2 \to x_3 \to \cdots \to x_n$$

- Interested in the marginal probability $p(x_n)$.

$$\boxed{O(k^{n-1})!}$$

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} p(x_1, \ldots, x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} p(x_1) \prod_{i=2}^{n} p(x_i \mid x_{i-1})$$

- Variable elimination by leveraging the factorization property.

$$\sum_{x_{n-1}} p(x_n \mid x_{n-1}) \sum_{x_{n-2}} p(x_{n-1} \mid x_{n-2}) \cdots \sum_{x_1} p(x_2 \mid x_1) p(x_1)$$

Depend on x_{n-1}, x_n

x_1 is eliminated

Example: Markov Chain

- Define **factors** as follows:

$$\sum_{x_{n-1}} p(x_n \mid x_{n-1}) \underbrace{\sum_{x_{n-2}} p(x_{n-1} \mid x_{n-2}) \cdots \underbrace{\sum_{x_1} p(x_2 \mid x_1) p(x_1)}_{\tau(x_2)}}_{\tau(x_{n-1})}$$

function MARGINAL-INFERENCE(*Markov chain*) **returns** $p(x_n)$
    for each assignment s to x_1 **do**
        $\tau(x_1 = s) = p(x_1 = s)$ (1). Initialization
    for t from 2 to n **do**
        for each assignment s to x_k **do**
            $\tau(x_t = s) = \sum_{x_{t-1}} p(x_t = s \mid x_{t-1}) \tau(x_{t-1})$ (2). Elimination
    return $p(x_n) = \tau(x_n)$.

$O(nk^2)$

Factors

- We are given a graphical model as a product of factors:

$$p(x_1, \dots, x_n) = \prod_{s \in F} \phi_s(\boldsymbol{x}_s).$$

Bayesian Networks	Markov Random Field
$p(\boldsymbol{x}) = \prod_{s=1}^{N} p(x_s \vert \boldsymbol{x}_{\Gamma(s)})$ $\Gamma(s)$ denotes parents of x_s.	$p(x) = \frac{1}{Z} \prod_{C} \phi_C(\boldsymbol{x}_C)$ $\phi_C(\boldsymbol{x}_C)$: functions of cliques
$F = \{s \cup \Gamma(s)\ \forall s\}$ $\phi_s = p(x_s \mid \boldsymbol{x}_{\Gamma(s)})$	F is the set of cliques $\phi_s = \phi_C(\boldsymbol{x}_C)/Z'$

Operations

- **Product**

$$\phi_3(x_c) = \phi_1(x_c^{(1)}) \times \phi_2(x_c^{(2)})$$

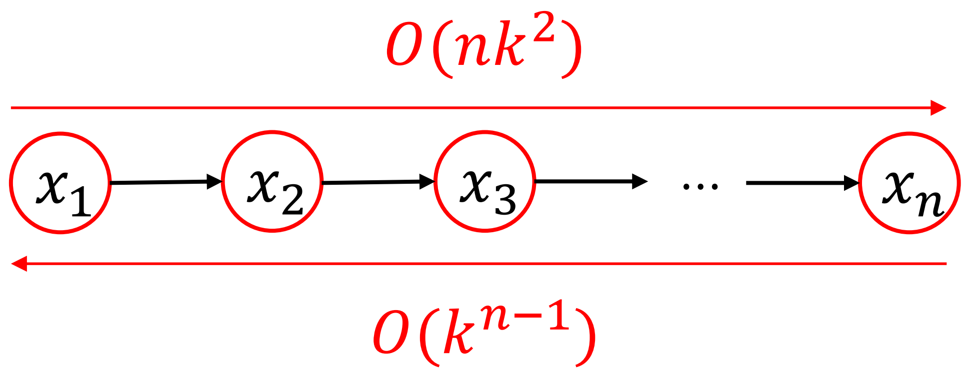
  - $x_c^{(i)}$ denotes an assignment to the variables in the scope of ϕ_i.
  - For example, $\phi_3(a, b, c) = \phi_1(a, b) \times \phi_2(b, c)$.

- **Marginalization**
  - If we have a factor $\phi(X, Y)$ over two sets of variables X, Y, marginalizing Y produces a new factor:

$$\tau(x) = \sum_y \phi(x, y)$$

Ordering

- In the example of Markov chain, we eliminate variables with the positive sequence ordering $(x_1 \to x_2 \to \cdots \to x_{n-1})$.
- Important notations:
  - Different orderings may dramatically alter the running time of the variable elimination algorithm.



  - It is NP-hard to find the best ordering.

Variable Elimination

function VARIABLE-ELIMINATION(*graphical model*) **returns** $p(x_n)$

  Initialize factors set Φ of the graphical model.

  Select an ordering O among variables except x_n. (1). Initialization

  for each variable x_i ordered according to O **do**

    $S = \{\phi_j \mid \phi_j \in \Phi, \phi_j \text{ depends on } x_i\}$ (2). Find relevant factors

    $p_i = \prod_{\phi_j \in S} \phi_j$

    $\tau = \sum_{x_i} p_i$ (3). Marginalization

    $\Phi = \Phi \cup \{\tau\} \setminus S$

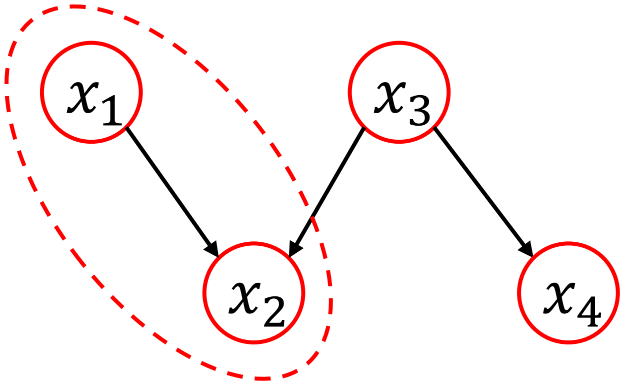
  return $p(x_n) = \tau(x_n)$.

- Available for both directed (Bayesian) and undirected (Markov) graphs.

Variable Elimination

- **Algorithm (Variable Elimination, VE)**
- For each variable X_i (ordered according to O):
  1. Multiply all factors ϕ_i containing X_i ;
  2. Marginalize out X_i to obtain a new factor τ ;
  3. Replace the factors ϕ_i with τ.

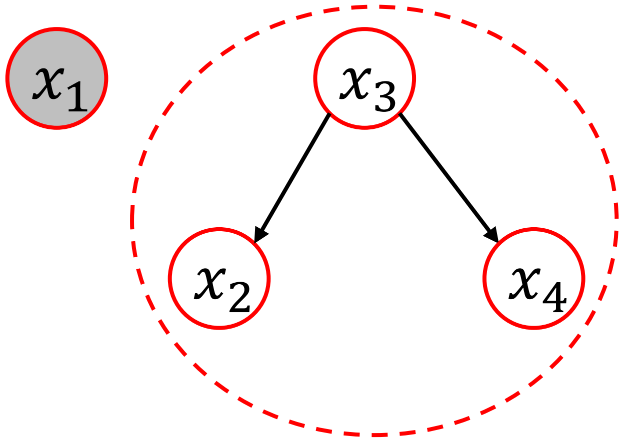
- Example: Alarm network
  - $\tau(x_2, x_3) = \sum_{x_1} p(x_2 \mid x_1, x_3) p(x_1)$



Variable Elimination

- **Algorithm (Variable Elimination, VE)**
- For each variable X_i (ordered according to O):
  1. Multiply all factors ϕ_i containing X_i ;
  2. Marginalize out X_i to obtain a new factor τ ;
  3. Replace the factors ϕ_i with τ.

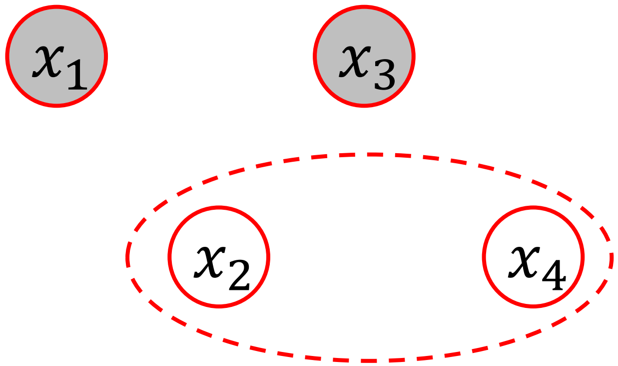
- Example: Alarm network
  - $\tau(x_2, x_3) = \sum_{x_1} p(x_2 \mid x_1, x_3) p(x_1)$
  - $\tau(x_2, x_4) = \sum_{x_3} \tau(x_2, x_3) p(x_3) p(x_4 \mid x_3)$



Variable Elimination

- **Algorithm (Variable Elimination, VE)**
- For each variable X_i (ordered according to O):
  1. Multiply all factors ϕ_i containing X_i ;
  2. Marginalize out X_i to obtain a new factor τ ;
  3. Replace the factors ϕ_i with τ.

- Example: Alarm network
  - $\tau(x_2, x_3) = \sum_{x_1} p(x_2 \mid x_1, x_3) p(x_1)$
  - $\tau(x_2, x_4) = \sum_{x_3} \tau(x_2, x_3) p(x_3) p(x_4 \mid x_3)$
  - $p(x_2) = \tau(x_2) = \sum_{x_4} \tau(x_2, x_4)$



Introducing Evidence

- Consider a general distribution $P(X, Y, E)$ over sets of:
  - Query variables Y;
  - Observed evidence variables E;
  - Unobserved variables X.

$$P(Y \mid E = e) = \left.\frac{P(Y, E = e)}{P(E = e)}\right\} \text{Apply VE Algorithm!}$$

- We can select the elimination ordering $X \rightarrow Y \rightarrow E$ to obtain $P(Y, E = e)$ and $P(E = e)$ in a single run of VE algorithm.
- How to reutilize the computation of $P(Y_1 \mid E_1 = e_1)$ for new query?
  - Example query: $P(Y_2 \mid E_2 = e_2)$.

Outline

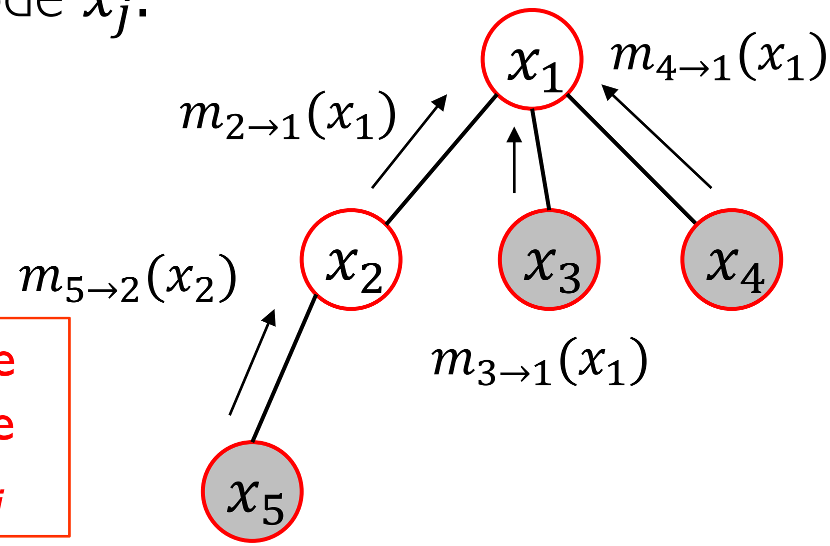
- Probabilistic Reasoning
- Graphical Models
  - Bayesian Networks
  - Markov Random Fields
- Conditional Independence
  - D-separation
  - Markov Properties
- **Inference in Graphical Models**
  - Variable Elimination
  - **Belief Propagation**

Variable Elimination on Tree

- We can find the optimal ordering by ranking the nodes in post-order.
  - A post-order traversal visits a node after its children.
- At each step, we will eliminate a leaf node x_j.
  - $\tau_k(x_k) = \sum_{x_j} \phi(x_k, x_j)\tau_j(x_j)$

Parent of x_j

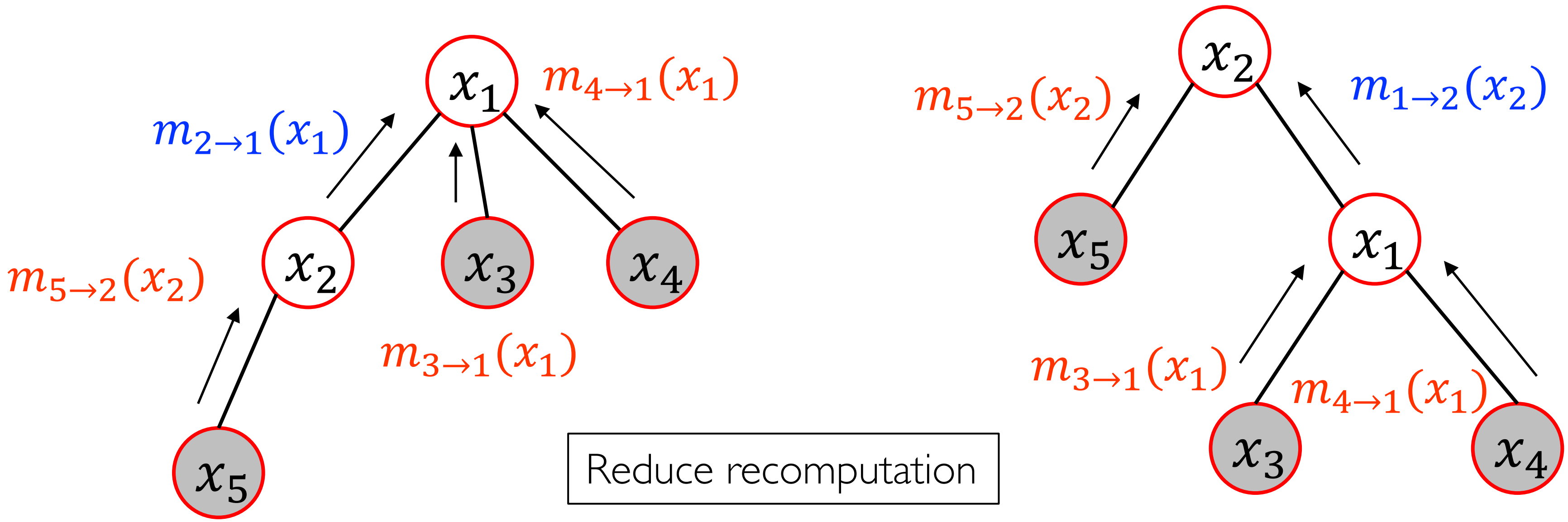
summarizes all of the information from the subtree rooted at x_j



  - $\tau_k(x_k)$ can be understood as the **massage** delivered from x_j to x_k, denoted as $m_{j\to k}(x_k)$

Variable Elimination on Tree

- We want to compute $p(x_2)$ after $p(x_1)$:



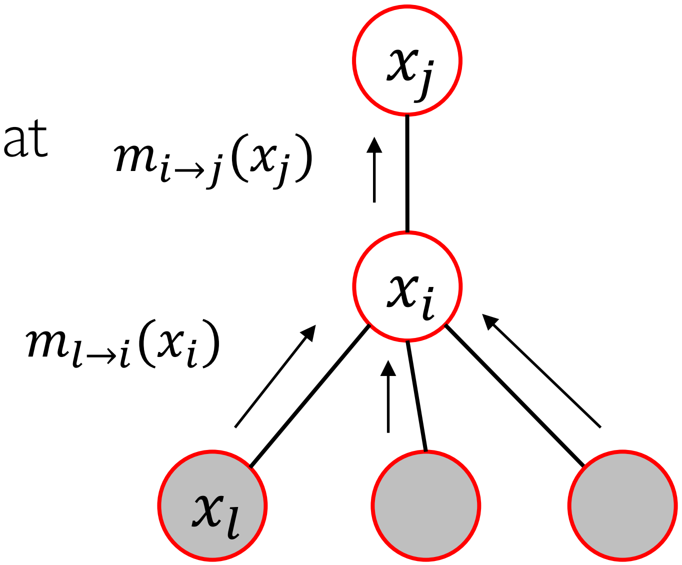
- Variable elimination processes **share messages**.
- Given a graph $G = (V, E)$, if we store $m_{i\to j}(x_j)$ and $m_{j\to i}(x_i)$ for each $(i,j) \in E$, we can compute $p(x_i)$ with $O(1)$ steps in average.

Sum-Product Massage Passing

- The message transmits from x_i to x_j:

$$m_{i\to j}(x_j) = \sum_{x_i} \phi(x_i)\phi(x_i, x_j) \prod_{l\in N(i)\backslash j} m_{l\to i}(x_i)$$

where $N(i)\backslash j$ refers to the set of nodes that are neighbors of i, excluding j.

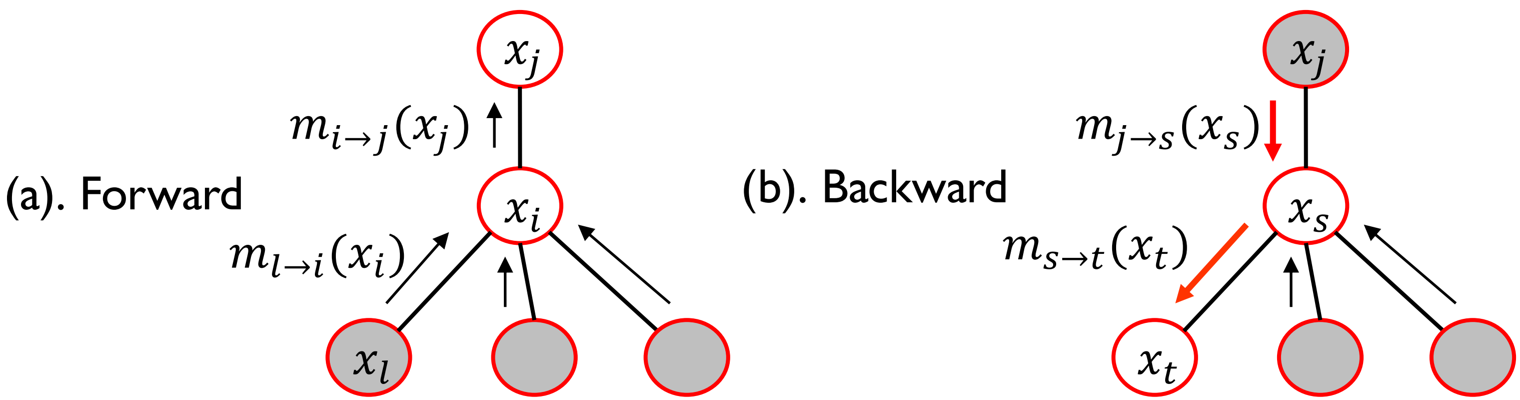


- After we have computed all messages, we may answer any marginal query over x_i.

$$p(x_i) \propto \phi(x_i) \prod_{l\in N(i)} m_{l\to i}(x_i)$$

Massage Passing Algorithm

1. **function** MASSAGE-PASSING(*tree model*) **returns** marginal functions
2. Pick an arbitrary node x_r as root.
3. Sort the nodes in post-order O.
4. **for** each non-root variable x_i ordered according to O **do**
5. Compute and propagate messages from x_i to its parent x_j. (a). Forward
6. **for** each non-root variable x_s ordered according to reverse O **do** (b). Backward
7. Compute and propagate messages from x_s to each of its children x_t
8. Compute the product of received messages at each node for x_n.



Message Passing for MAP Inference

- **Distributive law:** Sum and max operators distribute over products.
- Message passing can be used to perform MAP inference.
- Recall the example of Markov chain:

Marginal

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} p(x_1) \prod_{i=2}^{n} p(x_i \mid x_{i-1})$$

Σ

$$\sum_{x_{n-1}} p(x_n \mid x_{n-1}) \sum_{x_{n-2}} p(x_{n-1} \mid x_{n-2}) \cdots \sum_{x_1} p(x_2 \mid x_1) p(x_1)$$

MAP

$$\max_{x_1,\ldots,x_n} p(x_1, \ldots x_n) = \max_{x_1} \cdots \max_{x_n} p(x_1) \prod_{i=2}^{n} p(x_i \mid x_{i-1})$$

max

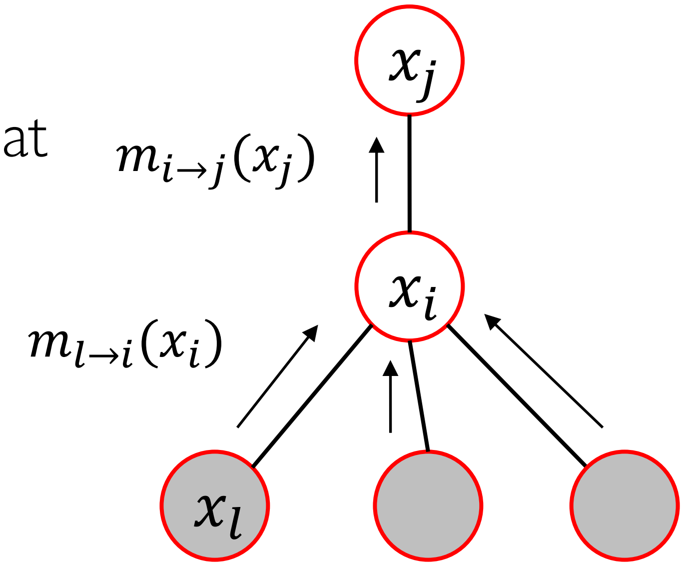
$$\max_{x_n} \max_{x_{n-1}} p(x_n \mid x_{n-1}) \cdots \max_{x_1} p(x_2 \mid x_1) p(x_1)$$

Max-Product Massage Passing

- The message transmit from x_i to x_j:

$$m_{i\to j}(x_j) = \max_{x_i} \phi(x_i)\phi(x_i, x_j) \prod_{l\in N(i)\backslash j} m_{l\to i}(x_i)$$

where $N(i)\backslash j$ refers to the set of nodes that are neighbors of i, excluding j.



- After we have computed all messages, we may answer the MAP inference.

$$p^* = \max_{x_i} \phi(x_i) \prod_{l\in N(i)} m_{l\to i}(x_i)$$

Loopy Belief Propagation

- Inference on general graphs (non-tree structure):
  - Approximate solution (other methods are in next lectures)

function MASSAGE-PASSING(*graphical model*) **returns** messages
    Initialize messages for each edge as uniform distribution.
    Select an ordering for edges O.
    loop until a fixed number of steps T or convergence **do**
        $t \leftarrow$ the number of iterations
        for each edge (i, j) ordered according to O **do**
            $m_{i \to j}^{t+1}(x_j) = \sum_{x_i} \phi(x_i)\phi(x_i, x_j) \prod_{l \in N(i) \setminus j} m_{l \to i}^{t}(x_i)$
            $m_{j \to i}^{t+1}(x_i) = \sum_{x_j} \phi(x_j)\phi(x_i, x_j) \prod_{l \in N(j) \setminus i} m_{l \to j}^{t}(x_j)$
    return messages $m_{i \to j}^{T}(x_j), m_{j \to i}^{T}(x_i)$ for each edge (i, j)

Thank You

Questions?

Mingsheng Long
mingsheng@tsinghua.edu.cn
http://ise.thss.tsinghua.edu.cn/~mlong

答疑：东主楼11区413室