RWTH AACHEN UNIVERSITY

Machine Learning – Lecture 14

Undirected Graphical Models & Inference

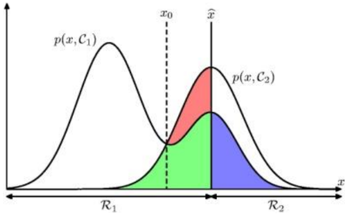
23.06.2015

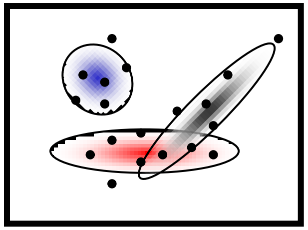
Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

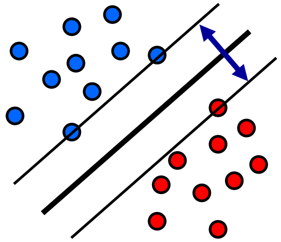
Many slides adapted from B. Schiele, S. Roth, Z. Gharahmani

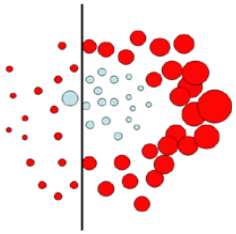
Course Outline



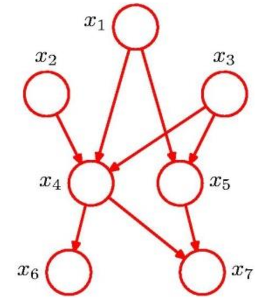


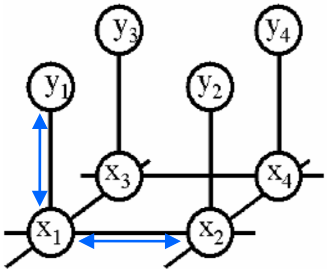
- Fundamentals (2 weeks)
  - Bayes Decision Theory
  - Probability Density Estimation





- Discriminative Approaches (5 weeks)
  - Linear Discriminant Functions
  - Statistical Learning Theory & SVMs
  - Ensemble Methods & Boosting
  - Decision Trees & Randomized Trees





- Generative Models (4 weeks)
  - Bayesian Networks
  - Markov Random Fields
  - Exact Inference

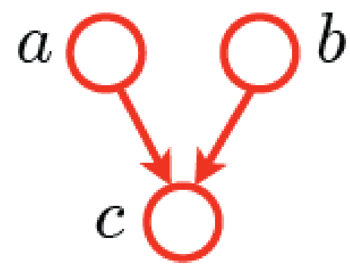
Topics of This Lecture

- Recap: Directed Graphical Models (Bayesian Networks)
  - Factorization properties
  - Conditional independence
  - Bayes Ball algorithm
- Undirected Graphical Models (Markov Random Fields)
  - Conditional Independence
  - Factorization
  - Example application: image restoration
  - Converting directed into undirected graphs
- Exact Inference in Graphical Models
  - Marginalization for undirected graphs
  - Inference on a chain
  - Inference on a tree
  - Message passing formalism

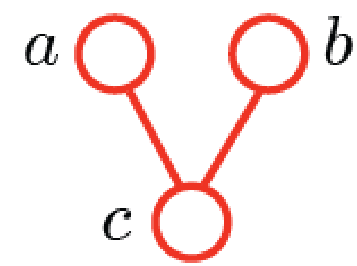
Recap: Graphical Models

- Two basic kinds of graphical models
  - Directed graphical models or Bayesian Networks
  - Undirected graphical models or Markov Random Fields

- Key components
  - Nodes
    - Random variables
  - Edges
    - Directed or undirected

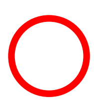


Directed graphical model

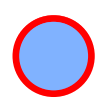


Undirected graphical model

  - The value of a random variable may be known or unknown.



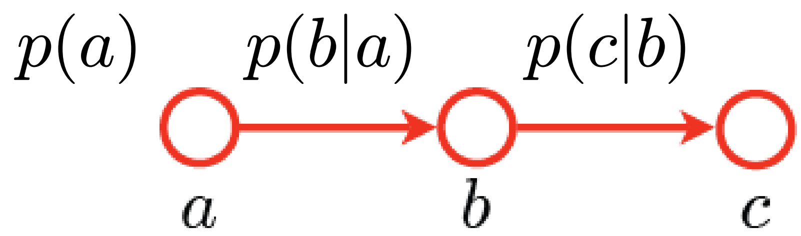
unknown



known

Recap: Directed Graphical Models

- **Chains of nodes:**



  - **Knowledge about a is expressed by the prior probability:**

$$p(a)$$

  - **Dependencies are expressed through conditional probabilities:**

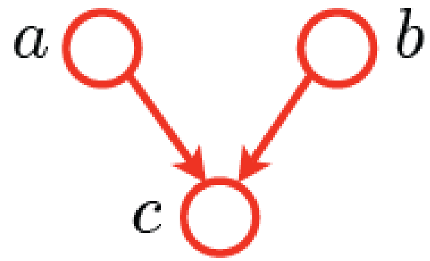
$$p(b|a), \;\; p(c|b)$$

  - **Joint distribution of all three variables:**

$$\begin{aligned} p(a,b,c) &= p(c|\cancel{a},b)p(a,b) \\ &= p(c|b)p(b|a)p(a) \end{aligned}$$

Recap: Directed Graphical Models

- **Convergent connections:**



  - **Here the value of c depends on both variables a and b.**
  - **This is modeled with the conditional probability:**

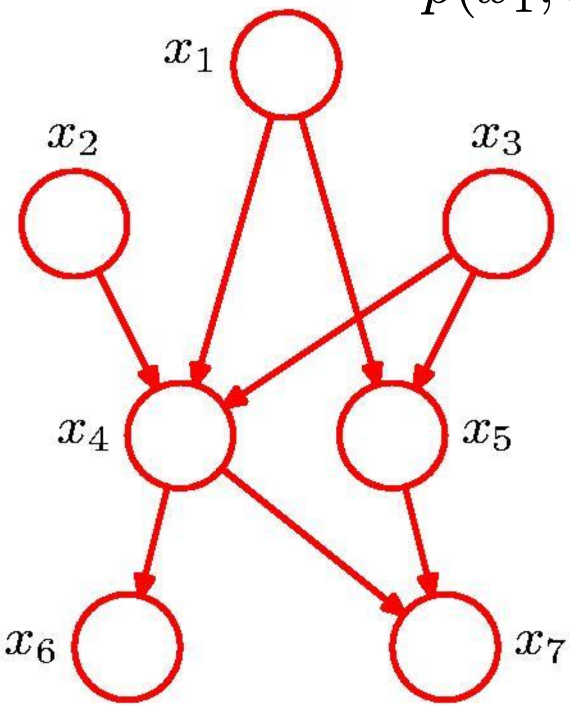
$$p(c|a, b)$$

  - **Therefore, the joint probability of all three variables is given as:**

$$\begin{aligned} p(a, b, c) &= p(c|a, b)p(a, b) \\ &= p(c|a, b)p(a)p(b) \end{aligned}$$

Recap: Factorization of the Joint Probability

- **Computing the joint probability**



$$p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \, p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

General factorization

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

We can directly read off the factorization of the joint from the network structure!

Image source: C. Bishop, 2006

Recap: Factorized Representation

- Reduction of complexity
  - Joint probability of n binary variables requires us to represent values by brute force

$$\mathcal{O}(2^n) \text{ terms}$$

  - The factorized form obtained from the graphical model only requires

$$\mathcal{O}(n \cdot 2^k) \text{ terms}$$

    – k: maximum number of parents of a node.

> ***⇒ It's the edges that are missing in the graph that are important! They encode the simplifying assumptions we make.***

Recap: Conditional Independence

- X is conditionally independent of Y given V
  - Definition: $X \perp\!\!\!\perp Y | V \quad \Leftrightarrow \quad p(X|Y,V) = p(X|V)$
  - Also: $X \perp\!\!\!\perp Y | V \quad \Leftrightarrow \quad p(X,Y|V) = p(X|V)\, p(Y|V)$
  - Special case: Marginal Independence

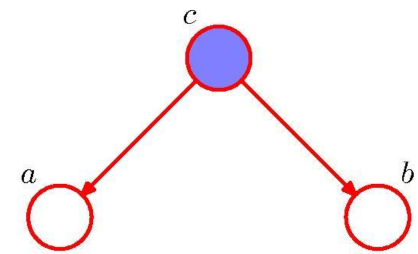
$$X \perp\!\!\!\perp Y \quad \Leftrightarrow \quad X \perp\!\!\!\perp Y | \emptyset \quad \Leftrightarrow \quad p(X,Y) = p(X)\, p(Y)$$

  - Often, we are interested in conditional independence between sets of variables:

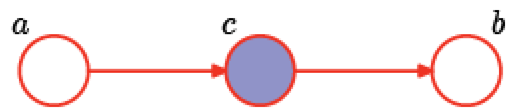
$$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{V} \quad \Leftrightarrow \quad \{X \perp\!\!\!\perp Y | \mathcal{V}, \;\; \forall X \in \mathcal{X} \;\; \text{and} \;\; \forall Y \in \mathcal{Y}\}$$

Recap: Conditional Independence

- Three cases
  - **Divergent** ("Tail-to-Tail")
    - Conditional independence when c is observed.



  - **Chain** ("Head-to-Tail")
    - Conditional independence when c is observed.



  - **Convergent** ("Head-to-Head")
    - Conditional independence when **neither** c, **nor any of its descendants** are observed.

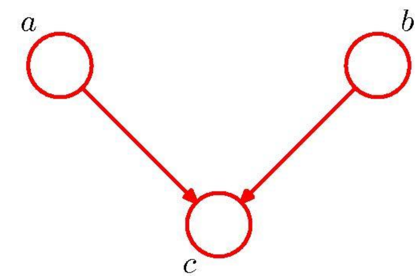
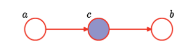
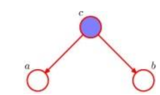


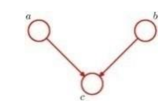
Image source: C. Bishop, 2006

Recap: D-Separation

- **Definition**
  - Let A, B, and C be non-intersecting subsets of nodes in a directed graph.
  - A path from A to B is **blocked** if it contains a node such that either
    - The arrows on the path meet either **head-to-tail** or **tail-to-tail** at the node, and the **node is in the set C**, or
    - The arrows meet **head-to-head** at the node, and **neither the node, nor any of its descendants, are in the set C**.
  - If all paths from A to B are blocked, A is said to be **d-separated** from B by C.

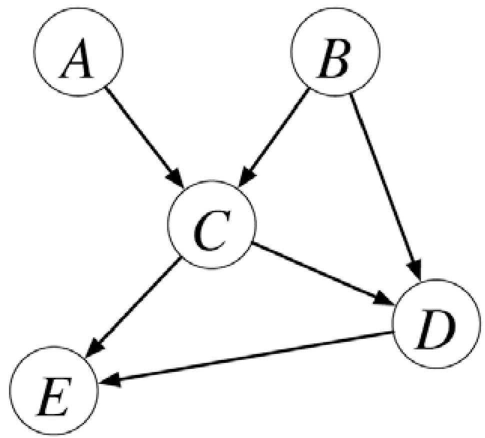






- **If A is d-separated from B by C, the joint distribution over all variables in the graph satisfies $A \perp\!\!\!\perp B \mid C$.**
  - Read: "A is conditionally independent of B given C."

Intuitive View: The "Bayes Ball" Algorithm



- **Game**
  - *Can you get a ball from X to Y without being blocked by $\mathcal{V}$?*
  - Depending on its direction and the previous node, the ball can
    - Pass through (from parent to all children, from child to all parents)
    - Bounce back (from any parent/child to all parents/children)
    - Be blocked

R.D. Shachter, Bayes-Ball: The Rational Pastime (for Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams), UAI'98, 1998

The "Bayes Ball" Algorithm

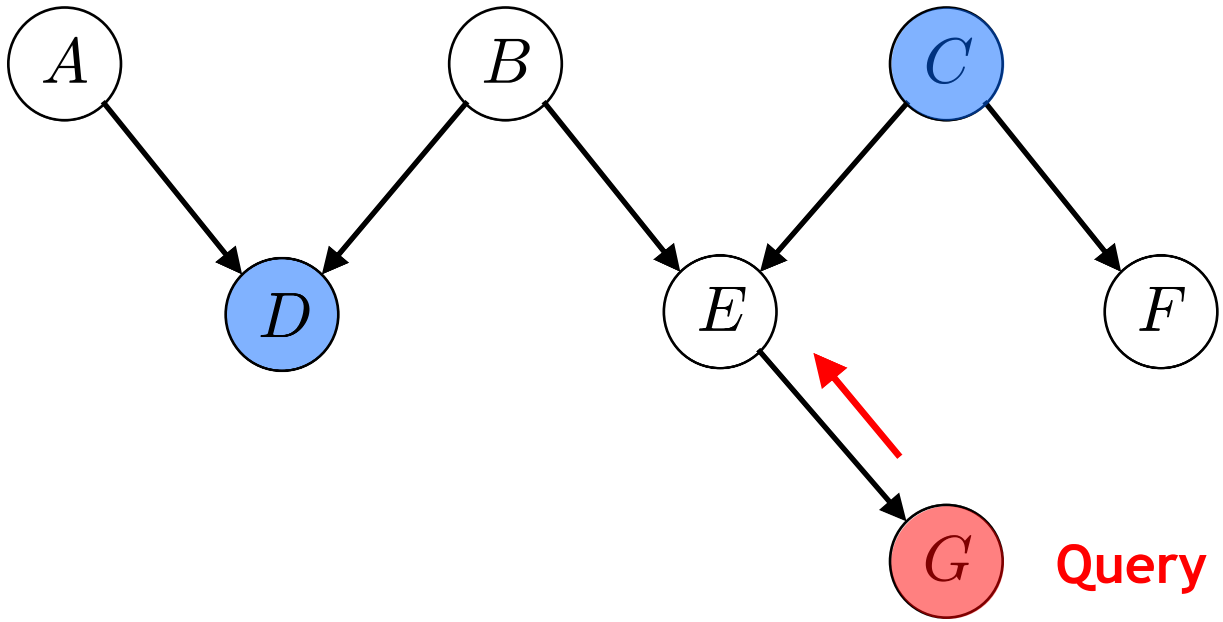
- Game rules
  - An unobserved node ($W \notin \mathcal{V}$) passes through balls from parents, but *also* bounces back balls from children.

  - An observed node ($W \in \mathcal{V}$) bounces back balls from parents, but blocks balls from children.

$\Rightarrow$ *The Bayes Ball algorithm determines those nodes that are d-separated from the query node.*

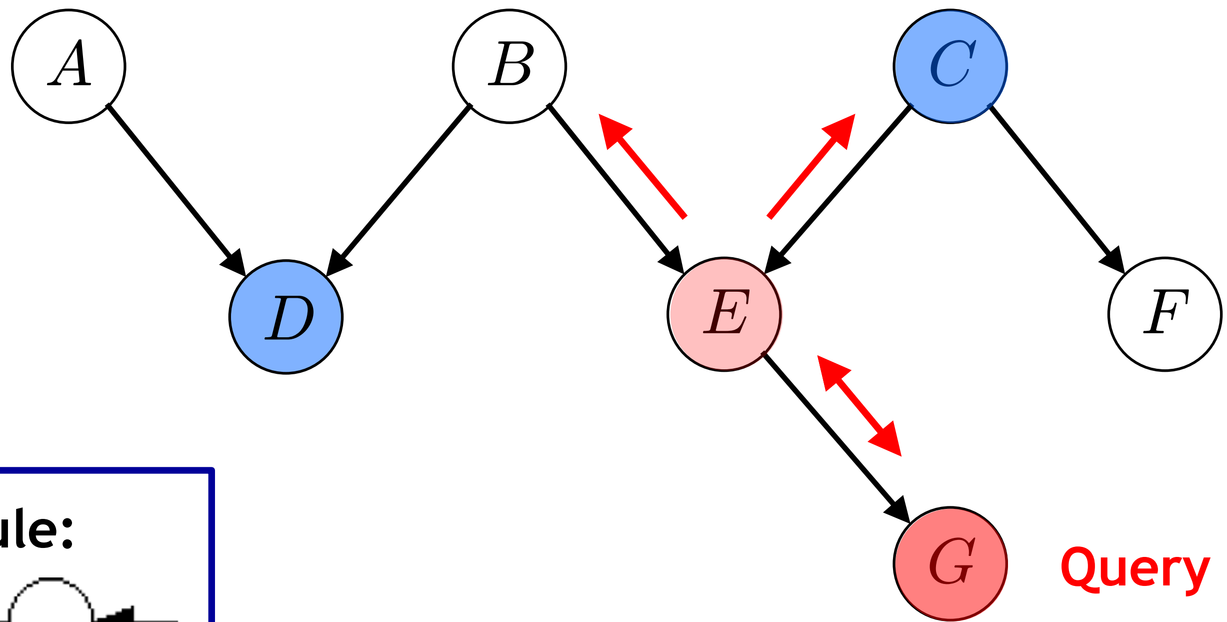
Image source: R. Shachter, 1998

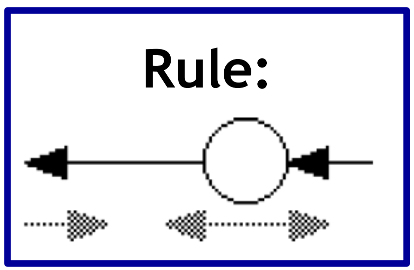
Example: Bayes Ball



- Which nodes are d-separated from G given C and D?

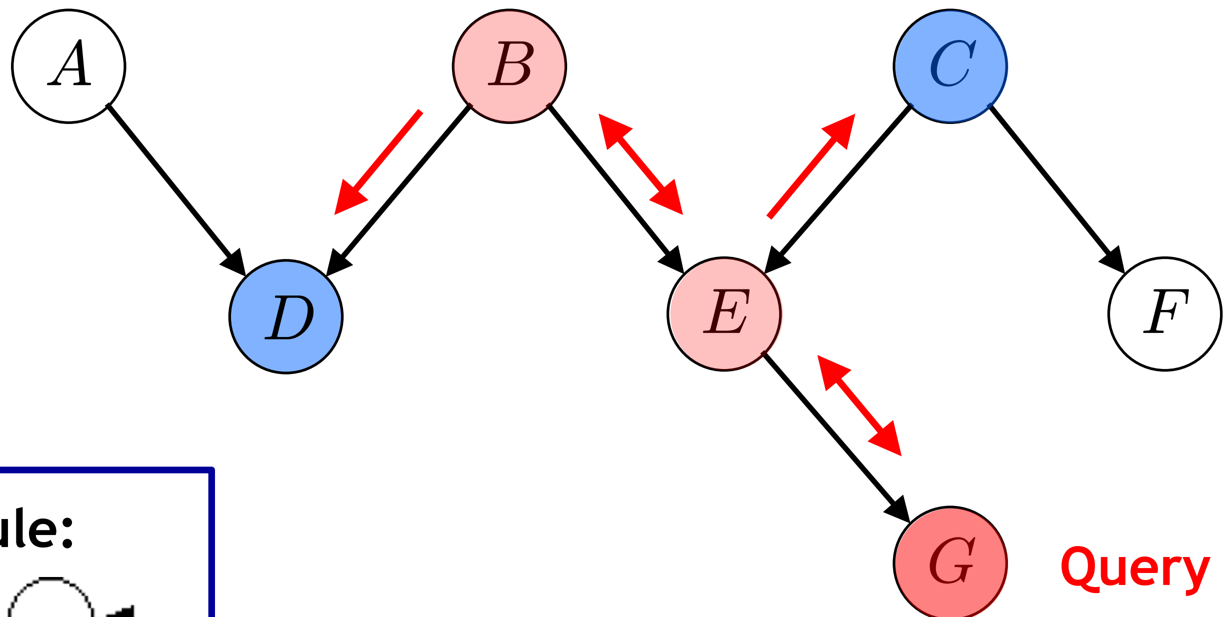
Example: Bayes Ball

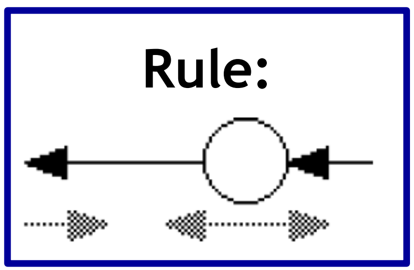




- Which nodes are d-separated from G given C and D?

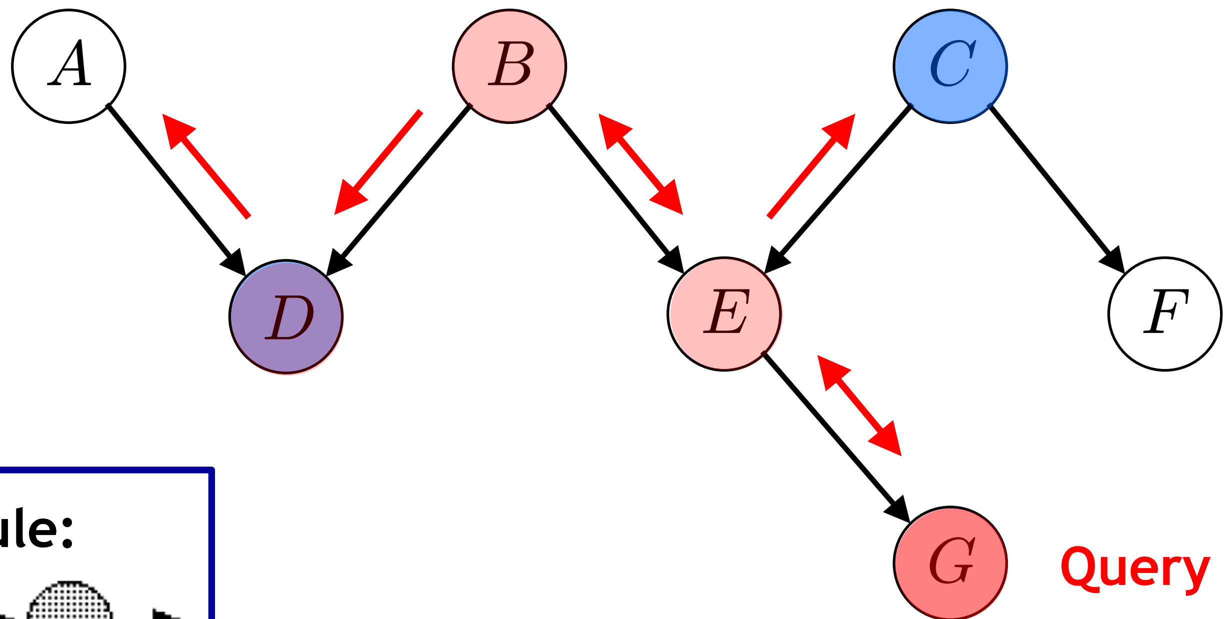
Example: Bayes Ball

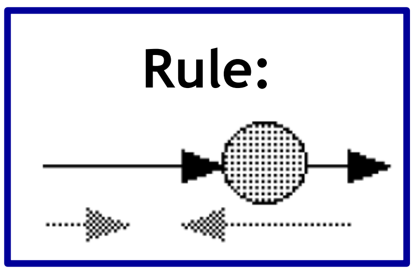




- Which nodes are d-separated from G given C and D?

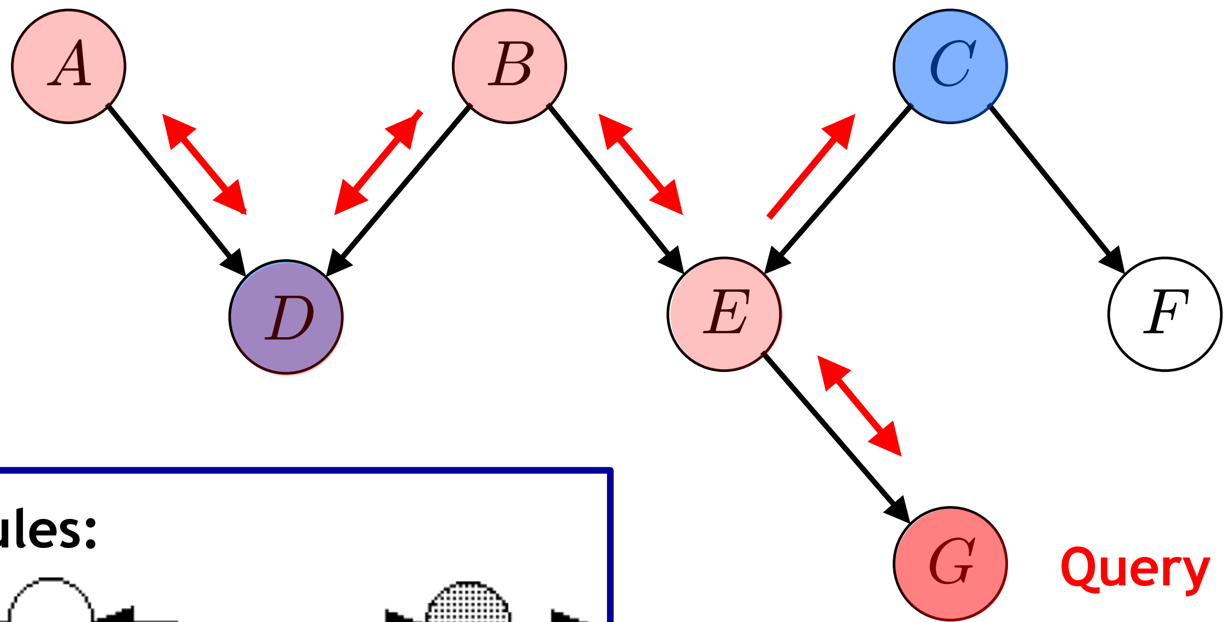
Example: Bayes Ball

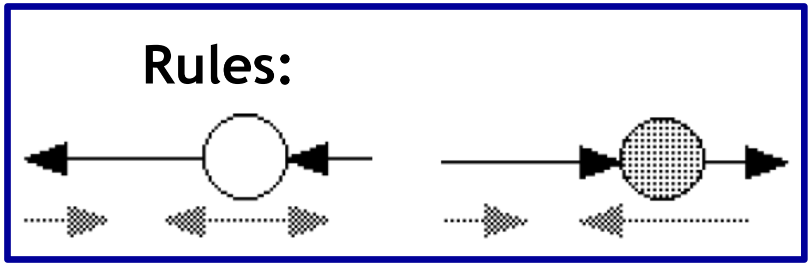




- Which nodes are d-separated from G given C and D?

Example: Bayes Ball

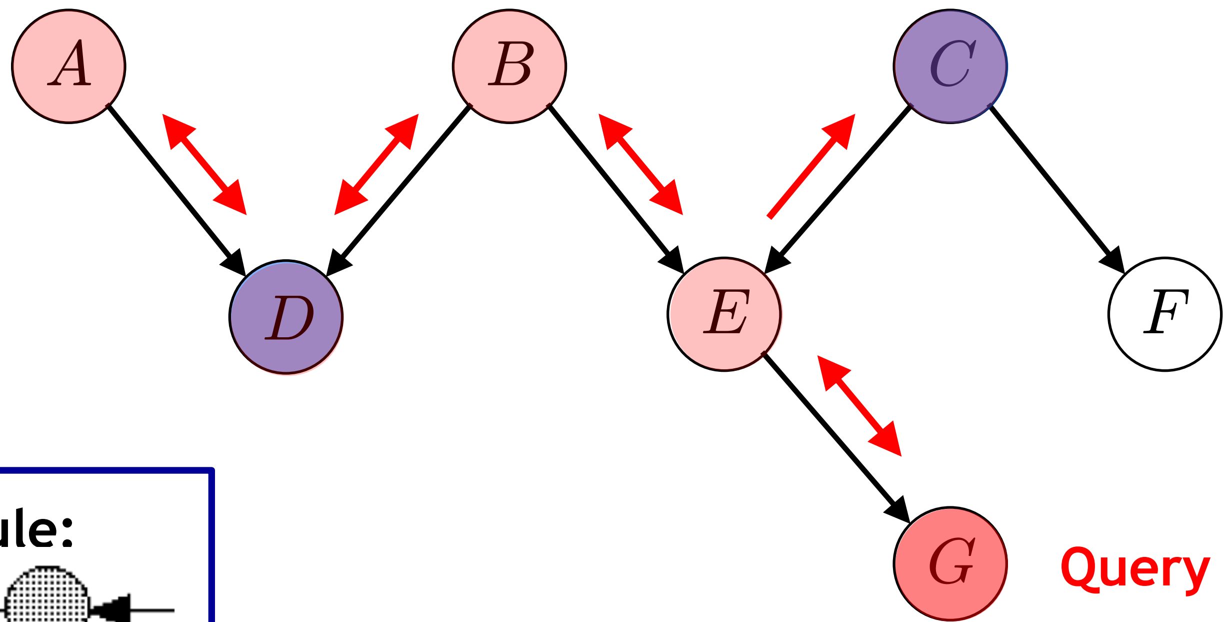


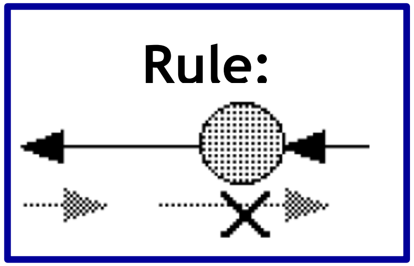


- Which nodes are d-separated from G given C and D?

RWTH AACHEN UNIVERSITY

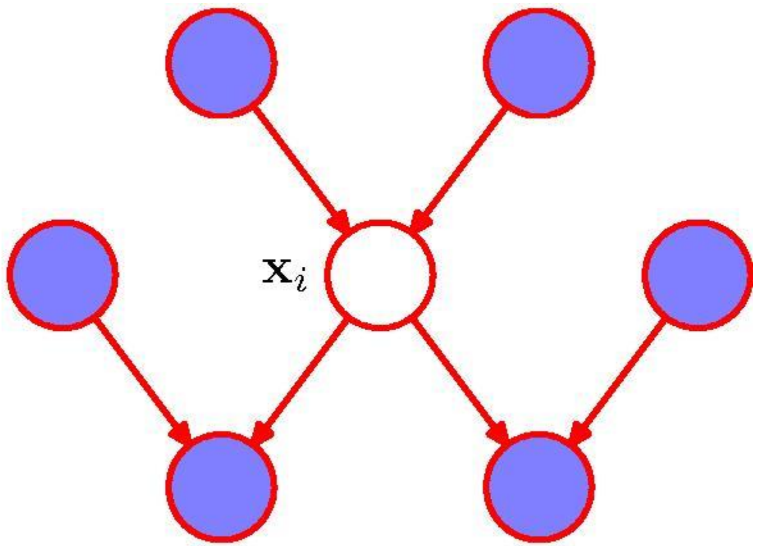
Example: Bayes Ball





- **Which nodes are d-separated from G given C and D?**
  - $\Rightarrow$ **F is d-separated from G given C and D.**

The Markov Blanket



- Markov blanket of a node $\mathbf{x}_i$
  - Minimal set of nodes that isolates $\mathbf{x}_i$ from the rest of the graph.
  - This comprises the set of
    - Parents,
    - Children, and
    - Co-parents of $\mathbf{x}_i$. ← **This is what we have to watch out for!**

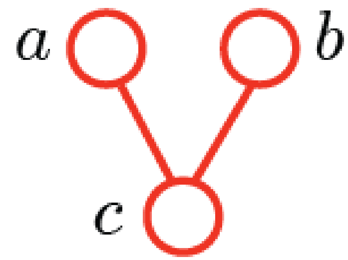
Image source: C. Bishop, 2006

Topics of This Lecture

- Recap: Directed Graphical Models (Bayesian Networks)
  - Factorization properties
  - Conditional independence
  - Bayes Ball algorithm
- **Undirected Graphical Models (Markov Random Fields)**
  - **Conditional Independence**
  - **Factorization**
  - **Example application: image restoration**
  - **Converting directed into undirected graphs**
- Exact Inference in Graphical Models
  - Marginalization for undirected graphs
  - Inference on a chain
  - Inference on a tree
  - Message passing formalism

Undirected Graphical Models

- Undirected graphical models ("Markov Random Fields")
  - Given by undirected graph



- Conditional independence is easier to read off for MRFs.
  - Without arrows, there is only one type of neighbors.
  - Simpler Markov blanket:

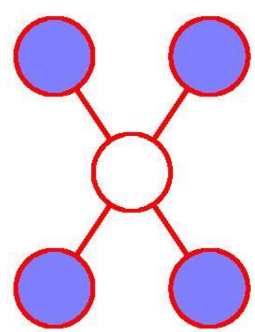
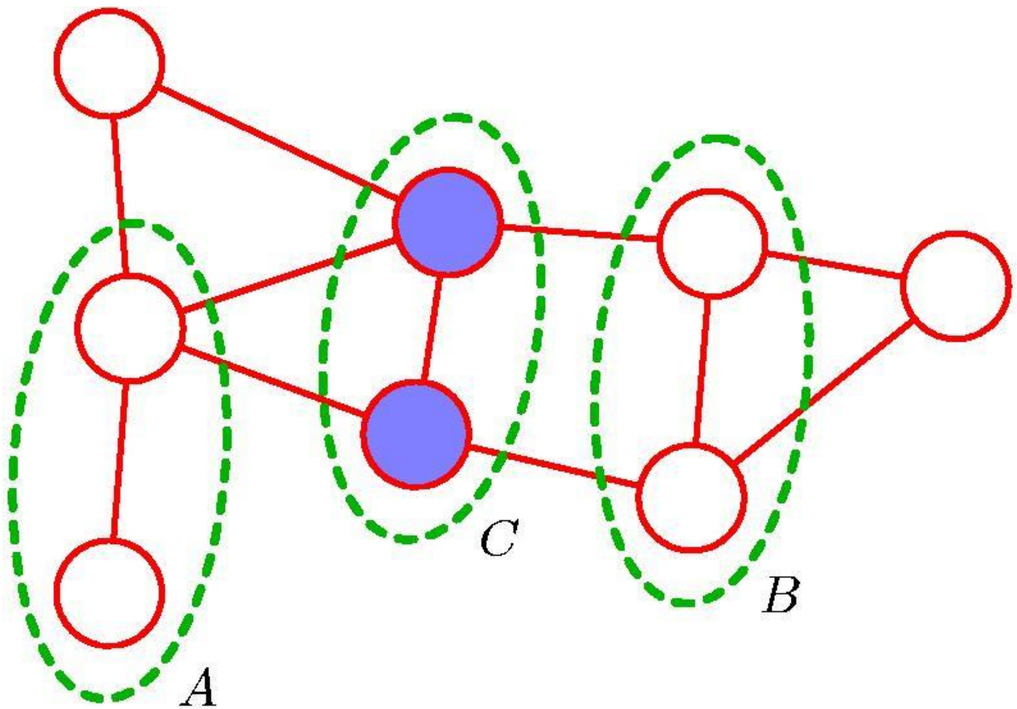


Image source: C. Bishop, 2006

Undirected Graphical Models



- **Conditional independence for undirected graphs**
  - **If every path from any node in set A to set B passes through at least one node in set C, then $A \perp\!\!\!\perp B \mid C$.**

Image source: C. Bishop, 2006

Factorization in MRFs

- Factorization
  - Factorization is more complicated in MRFs than in BNs.
  - Important concept: maximal cliques

  - Clique
    - Subset of the nodes such that there exists a link between all pairs of nodes in the subset.

  - Maximal clique
    - The biggest possible such clique in a given graph.

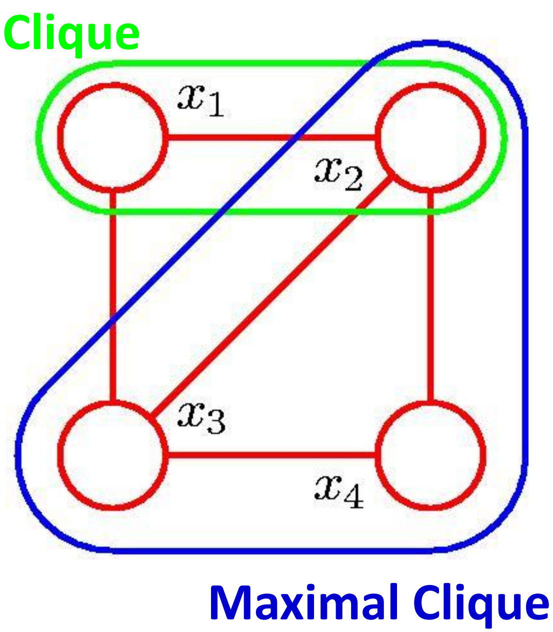


Image source: C. Bishop, 2006

Factorization in MRFs

- Joint distribution
  - Written as product of potential functions over maximal cliques in the graph:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$$

  - The normalization constant Z is called the partition function.

$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$$

- Remarks
  - BNs are automatically normalized. But for MRFs, we have to explicitly perform the normalization.
  - Presence of normalization constant is major limitation!
    - Evaluation of Z involves summing over $\mathcal{O}(K^M)$ terms for M nodes.

Factorization in MRFs

- **Role of the potential functions**
  - General interpretation
    - No restriction to potential functions that have a specific probabilistic interpretation as marginals or conditional distributions.
  - Convenient to express them as exponential functions ("Boltzmann distribution")

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

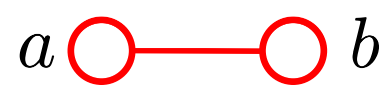
    - with an energy function E.
  - Why is this convenient?
    - Joint distribution is the product of potentials ⇒ sum of energies.
    - We can take the log and simply work with the sums…

Comparison: Directed vs. Undirected Graphs

- **Directed graphs** (Bayesian networks)
  - Better at expressing causal relationships.
  - Interpretation of a link:
    - Conditional probability $p(b|a)$.

  - Factorization is simple (and result is automatically normalized).
  - Conditional independence is more complicated.

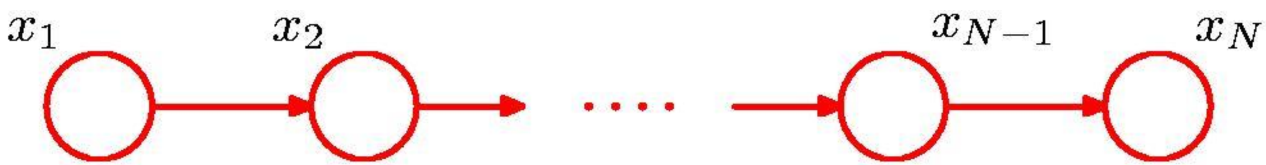
- **Undirected graphs** (Markov Random Fields)
  - Better at representing soft constraints between variables.
  - Interpretation of a link:
    - "There is some relationship between a and b".



  - Factorization is complicated (and result needs normalization).
  - Conditional independence is simple.

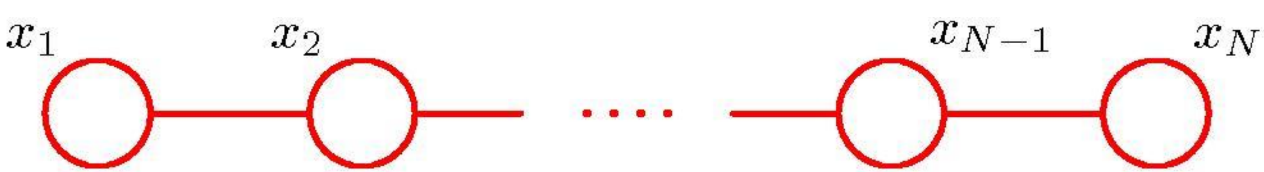
Converting Directed to Undirected Graphs

- **Simple case: chain**



$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)\,p(x_3|x_2)\cdots p(x_N|x_{N-1})$$

$$p(\mathbf{x}) = \frac{1}{Z}\,\psi_{1,2}(x_1,x_2)\,\psi_{2,3}(x_2,x_3)\cdots\psi_{N-1,N}(x_{N-1},x_N)$$

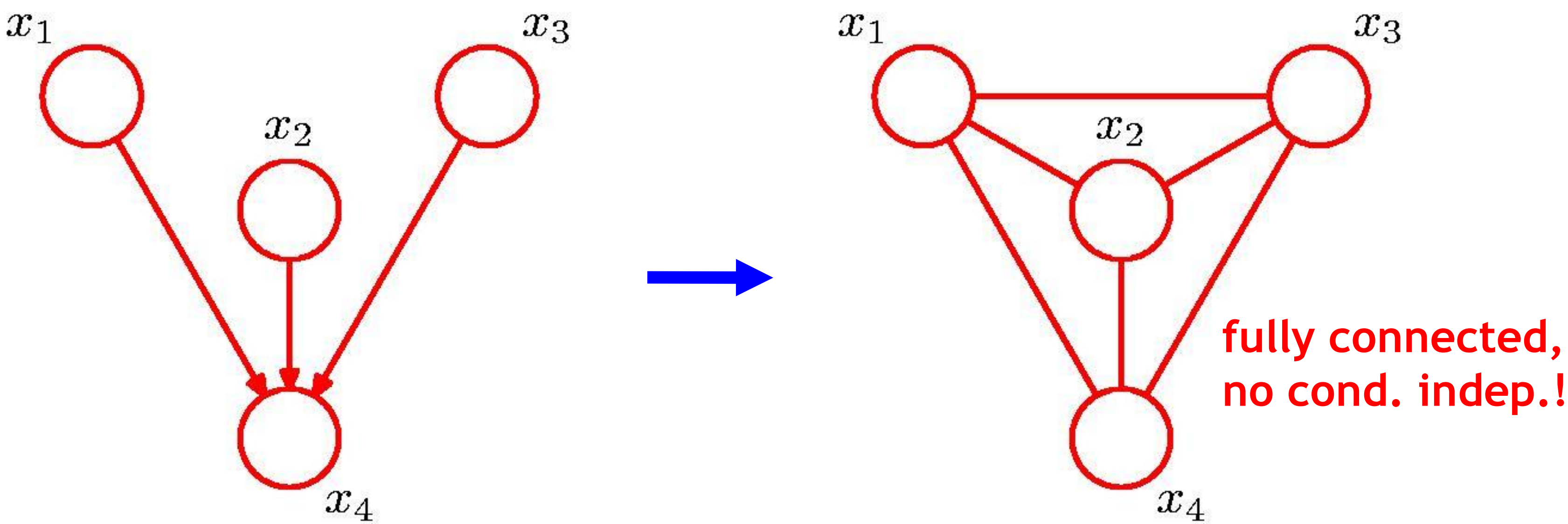


$\Rightarrow$ **We can directly replace the directed links by undirected ones.**

Image source: C. Bishop, 2006

Converting Directed to Undirected Graphs

- More difficult case: multiple parents



$$p(\mathbf{x}) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)$$

Need a clique of $x_1,\ldots,x_4$ to represent this factor!

- **Need to introduce additional links ("marry the parents").**

⇒ This process is called moralization. It results in the moral graph.

Image source: C. Bishop, 2006

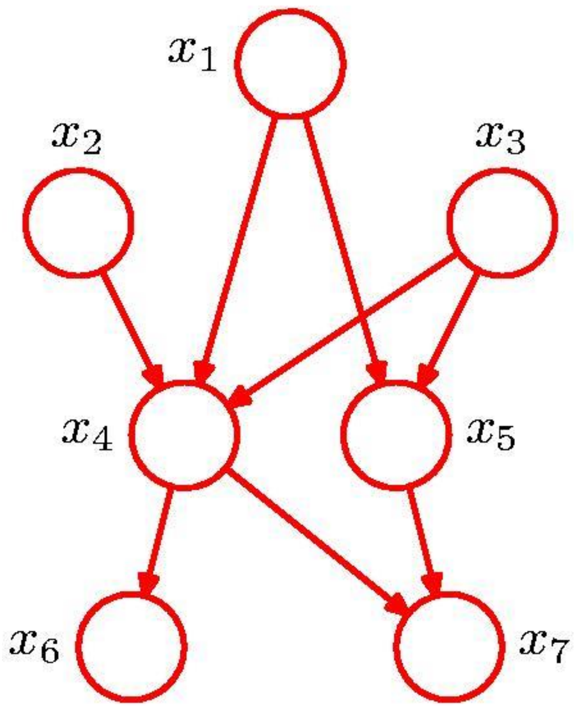
Converting Directed to Undirected Graphs

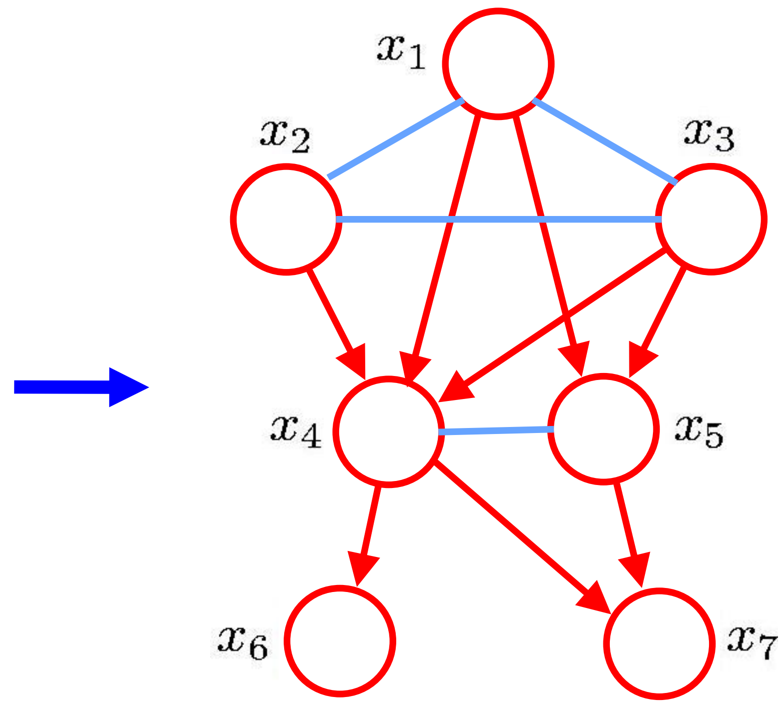
- General procedure to convert directed $\rightarrow$ undirected
  1. Add undirected links to marry the parents of each node.
  2. Drop the arrows on the original links $\Rightarrow$ moral graph.
  3. Find maximal cliques for each node and initialize all clique potentials to 1.
  4. Take each conditional distribution factor of the original directed graph and multiply it into one clique potential.

- Restriction
  - Conditional independence properties are often lost!
  - Moralization results in additional connections and larger cliques.

Example: Graph Conversion

- Step 1) Marrying the parents.

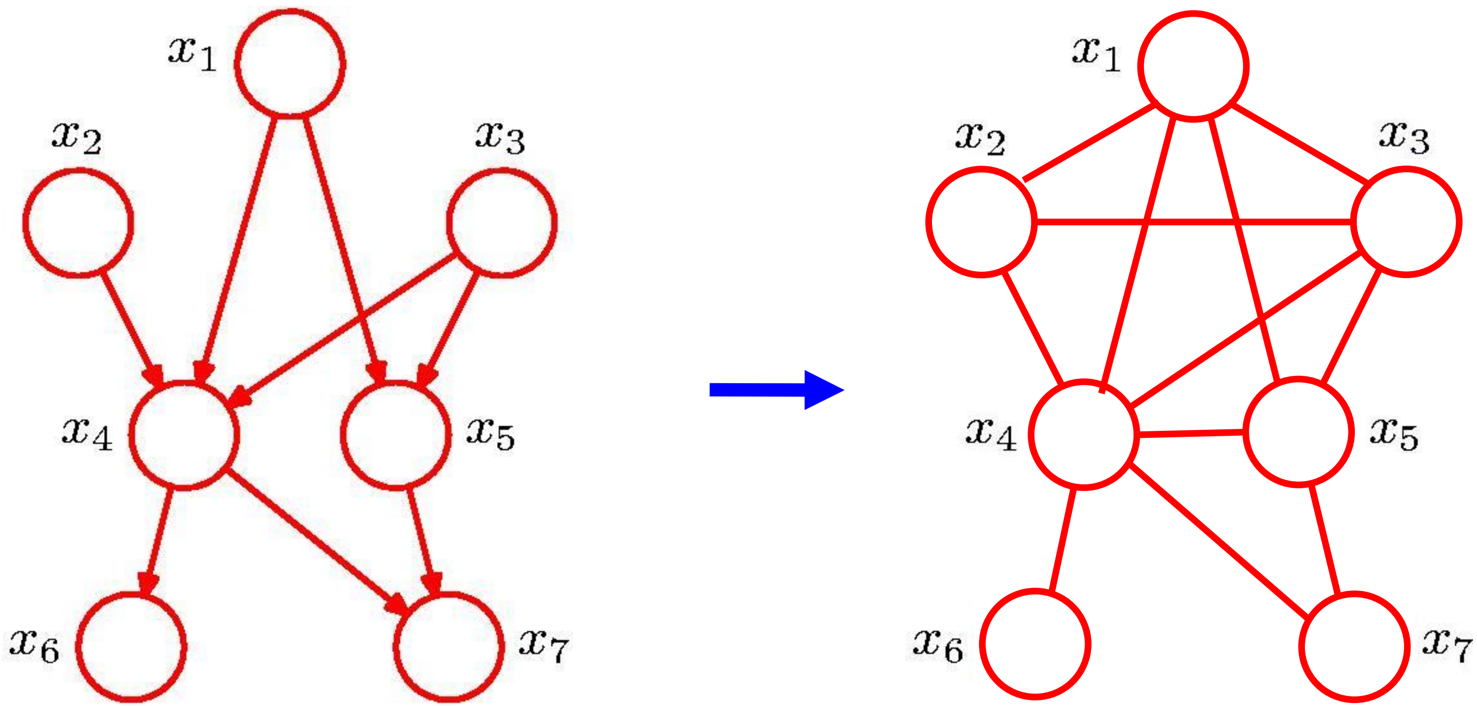




$$p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)$$
$$p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

Example: Graph Conversion

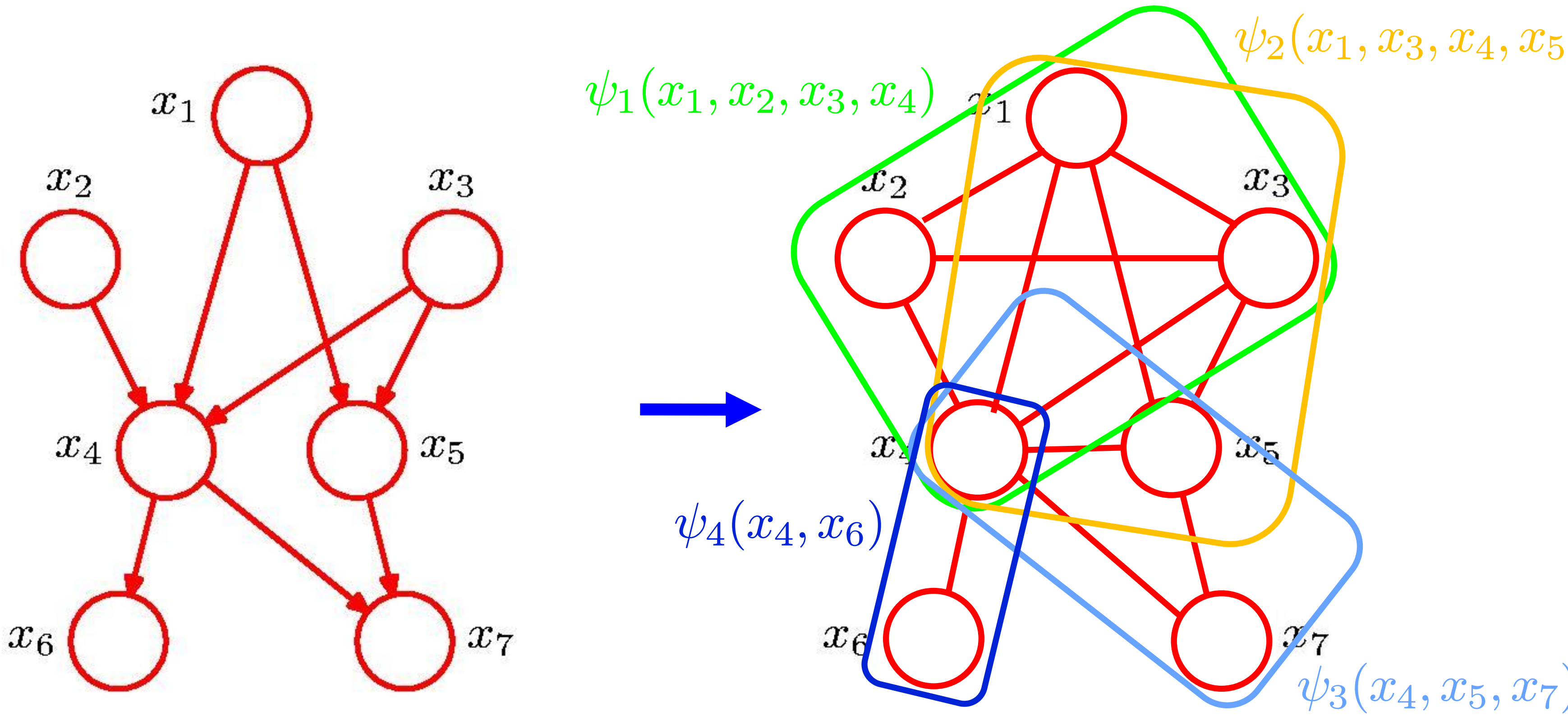
- **Step 2)** **Dropping the arrows.**



$$p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\ p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

Example: Graph Conversion

- **Step 3) Finding maximal cliques for each node.**



$$p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)$$
$$p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

RWTH AACHEN UNIVERSITY

Example: Graph Conversion

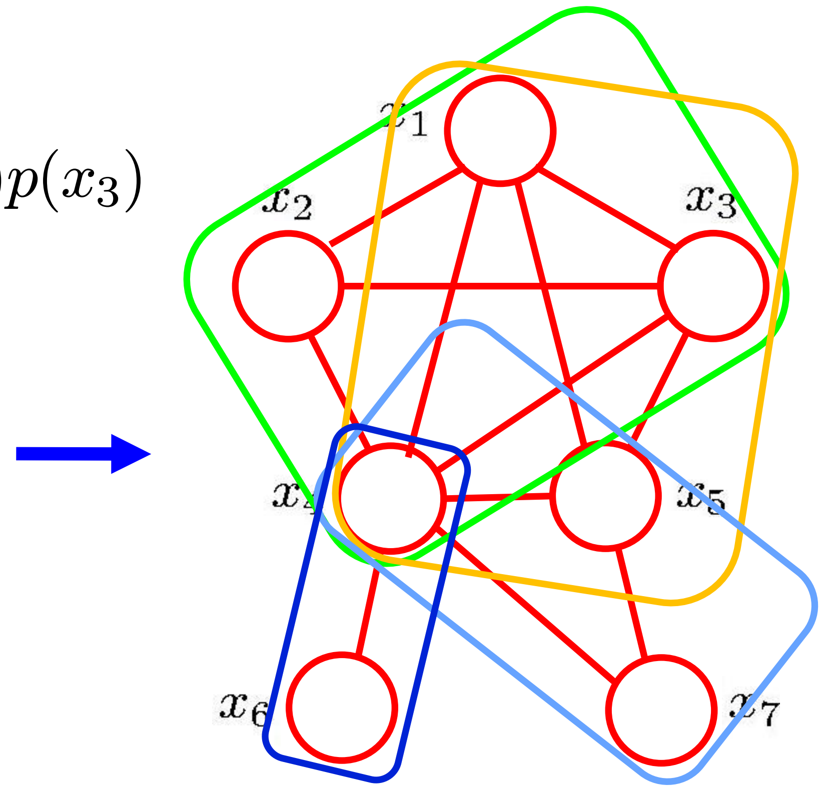
- **Step 4) Assigning the probabilities to clique potentials.**

$\psi_1(x_1, x_2, x_3, x_4)$
$= 1\ p(x_4|x_1, x_2, x_3)\, p(x_2) p(x_3)$

$\psi_2(x_1, x_3, x_4, x_5)$
$= 1\ p(x_5|x_1, x_3)\, p(x_1)$

$\psi_3(x_4, x_5, x_7)$
$= 1\ p(x_7|x_4, x_5)$

$\psi_4(x_4, x_6)$
$= 1\ p(x_6|x_4)$

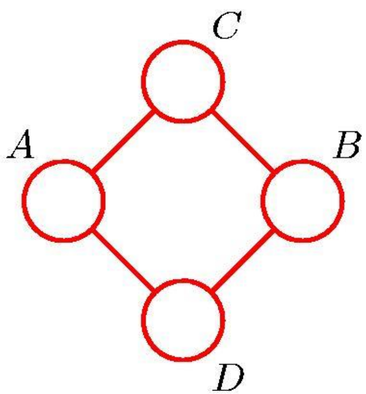


$$p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\ p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

Comparison of Expressive Power

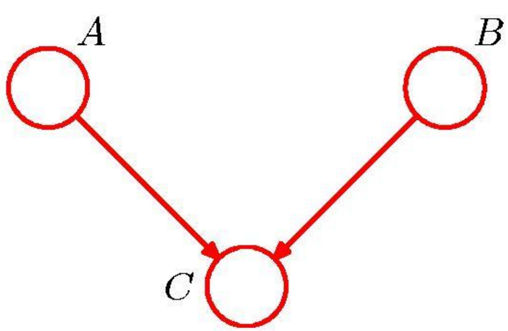
- **Both types of graphs have unique configurations.**

$$A \not\perp\!\!\!\perp B \mid \emptyset$$
$$A \perp\!\!\!\perp B \mid C \cup D$$
$$C \perp\!\!\!\perp D \mid A \cup B$$



No directed graph can represent these and only these independencies.

$$A \perp\!\!\!\perp B \mid \emptyset$$
$$A \not\perp\!\!\!\perp B \mid C$$



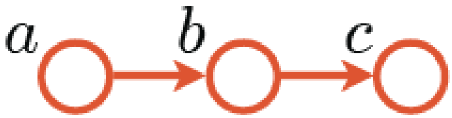
No undirected graph can represent these and only these independencies.

Topics of This Lecture

- Recap: Directed Graphical Models (Bayesian Networks)
  - Factorization properties
  - Conditional independence
  - Bayes Ball algorithm
- Undirected Graphical Models (Markov Random Fields)
  - Conditional Independence
  - Factorization
  - Example application: image restoration
  - Converting directed into undirected graphs
- **Exact Inference in Graphical Models**
  - **Marginalization for undirected graphs**
  - **Inference on a chain**
  - **Inference on a tree**
  - **Message passing formalism**

Inference in Graphical Models

- **Example 1:**

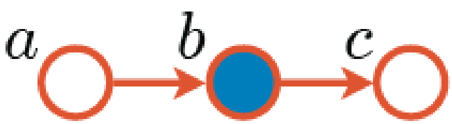


Goal: compute the marginals

$$p(a) = \sum_{b,c} p(a)p(b|a)p(c|b)$$

$$p(b) = \sum_{a,c} p(a)p(b|a)p(c|b)$$

- **Example 2:**



$$p(a|b=b') = \sum_{c} p(a)p(b=b'|a)p(c|b=b')$$

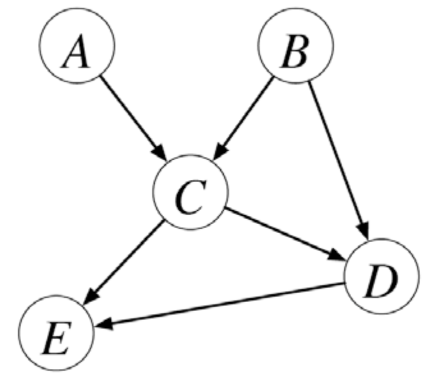
$$= p(a)p(b=b'|a)$$

$$p(c|b=b') = \sum_{a} p(a)p(b=b'|a)p(c|b=b')$$

$$= p(c|b=b')$$

Inference in Graphical Models

- **Inference - General definition**
  - Evaluate the probability distribution over some set of variables, given the values of another set of variables (=observations).



- **Example:**

$$p(A, B, C, D, E) = p(A)p(B)p(C|A, B)p(D|B, C)p(E|C, D)$$

  - How can we compute $p(A|C = c)$?
  - Idea:

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

Inference in Graphical Models

- **Computing $p(A|C=c)$...**
  - We know
    $$p(A,B,C,D,E) = p(A)p(B)p(C|A,B)p(D|B,C)p(E|C,D)$$
  - Assume each variable is binary.

- **Naïve approach:**

Two possible values for each $\Rightarrow 2^4$ terms

$$p(A, C=c) = \sum_{B,D,E} p(A,B,C=c,D,E)$$ **16 operations**

$$p(C=c) = \sum_{A} p(A,C=c)$$ **2 operations**

$$p(A|C=c) = \frac{p(A,C=c)}{p(C=c)}$$ **2 operations**

Total: 16+2+2 = 20 operations

Inference in Graphical Models

- We know

$$p(A,B,C,D,E) = p(A)p(B)p(C|A,B)p(D|B,C)p(E|C,D)$$

- **More efficient method for $p(A|C=c)$:**

$$p(A,C=c) = \sum_{B,D,E} p(A)p(B)p(C=c|A,B)p(D|B,C=c)p(E|C=c,D)$$

$$= \sum_{B} p(A)p(B)p(C=c|A,B) \underbrace{\sum_{D} p(D|B,C=c)}_{=1} \underbrace{\sum_{E} p(E|C=c,D)}_{=1}$$

$$= \sum_{B} p(A)p(B)p(C=c|A,B)$$

4 operations

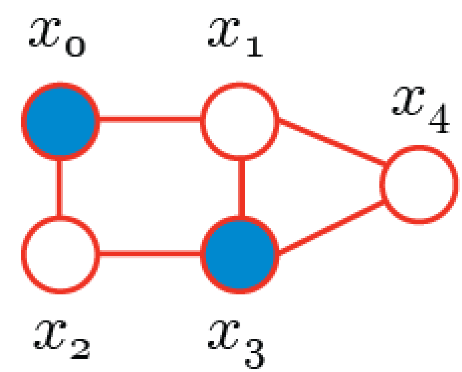
- Rest stays the same:

Total: 4+2+2 = 8 operations

- Strategy: Use the conditional independence in a graph to perform efficient inference.
  - ⇒ For singly connected graphs, exponential gains in efficiency!

Computing Marginals

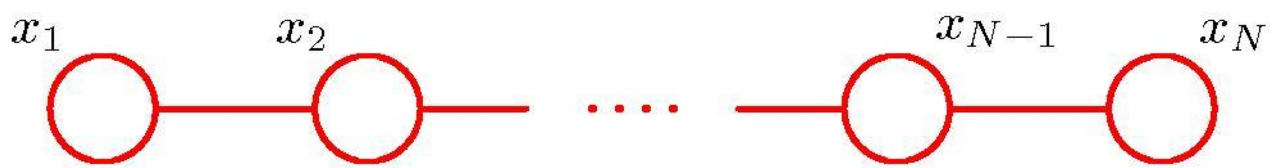
- How do we apply graphical models?
  - Given some observed variables, we want to compute distributions of the unobserved variables.
  - In particular, we want to compute marginal distributions, for example $p(x_4)$.



- How can we compute marginals?
  - Classical technique: sum-product algorithm by Judea Pearl.
  - In the context of (loopy) undirected models, this is also called (loopy) belief propagation [Weiss, 1997].
  - Basic idea: message-passing.

Inference on a Chain

- **Chain graph**



  - **Joint probability**

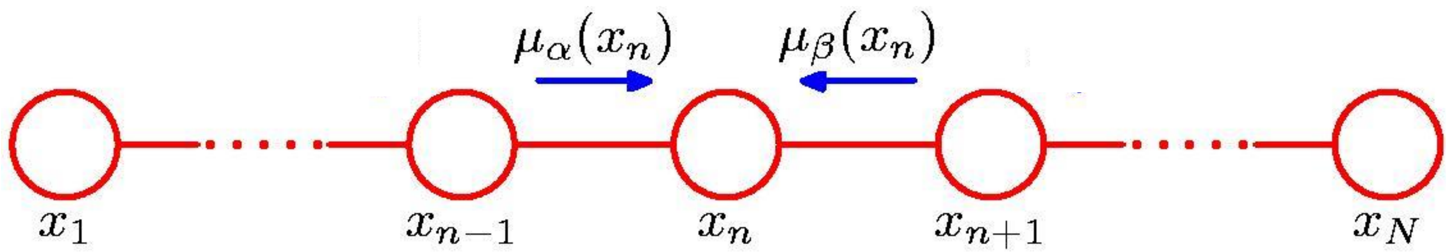
$$p(\mathbf{x}) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\cdots\psi_{N-1,N}(x_{N-1}, x_N)$$

  - **Marginalization**

$$p(x_n) = \sum_{x_1}\cdots\sum_{x_{n-1}}\sum_{x_{n+1}}\cdots\sum_{x_N} p(\mathbf{x})$$

Image source: C. Bishop, 2006

Inference on a Chain

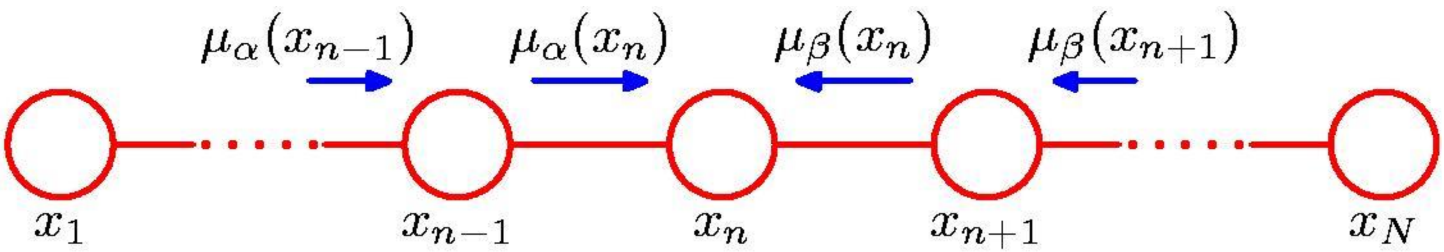


- **Idea: Split the computation into two parts ("messages").**

$$p(x_n) = \frac{1}{Z} \underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_1} \psi_{1,2}(x_1, x_2) \right] \cdots \right]}_{\mu_\alpha(x_n)}$$

$$\underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \right] \cdots \right]}_{\mu_\beta(x_n)}$$

Inference on a Chain



- **We can define the messages recursively…**

$$\mu_\alpha(x_n) = \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \left[\sum_{x_{n-2}} \cdots \right]$$

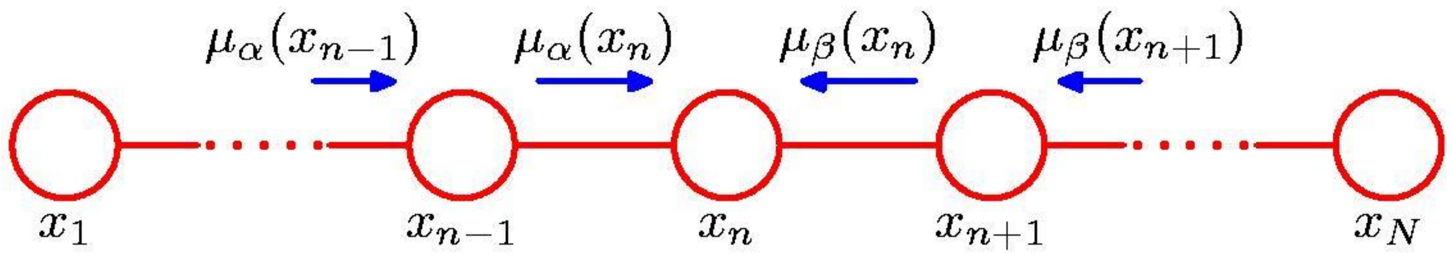
$$= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \mu_\alpha(x_{n-1}).$$

$$\mu_\beta(x_n) = \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \left[\sum_{x_{n+2}} \cdots \right]$$

$$= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \mu_\beta(x_{n+1}).$$

Image source: C. Bishop, 2006

Inference on a Chain



- **Until we reach the leaf nodes…**

$$\mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \qquad \mu_\beta(x_{N-1}) = \sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)$$

- **Interpretation**
  - **We pass messages from the two ends towards the query node x_n.**
- **We still need the normalization constant Z.**
  - **This can be easily obtained from the marginals:**

$$Z = \sum_{x_n} \mu_\alpha(x_n)\mu_\beta(x_n)$$

Image source: C. Bishop, 2006

Summary: Inference on a Chain

- To compute local marginals:
  - Compute and store all forward messages $\mu_\alpha(x_n)$.
  - Compute and store all backward messages $\mu_\beta(x_n)$.
  - Compute Z at any node x_m.
  - Compute

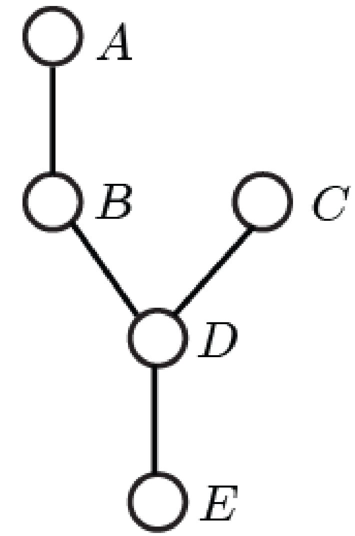
$$p(x_n) = \frac{1}{Z}\mu_\alpha(x_n)\mu_\beta(x_n)$$

    for all variables required.

- Inference through message passing
  - We have thus seen a first message passing algorithm.
  - How can we generalize this?

Inference on Trees

- Let's next assume a tree graph.
  - Example:



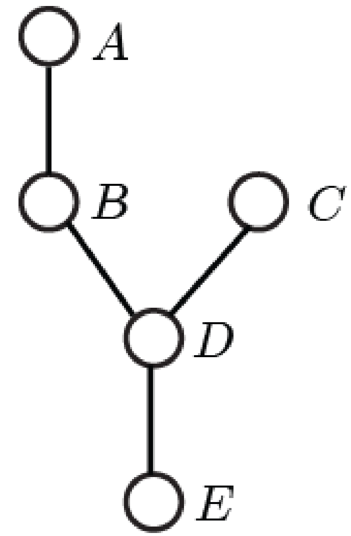
  - We are given the following joint distribution:

$$p(A,B,C,D,E) = \frac{1}{Z} f_1(A,B) \cdot f_2(B,D) \cdot f_3(C,D) \cdot f_4(D,E)$$

  - Assume we want to know the marginal $p(E)$...

Inference on Trees

- **Strategy**
  - **Marginalize out all other variables by summing over them.**



  - **Then rearrange terms:**

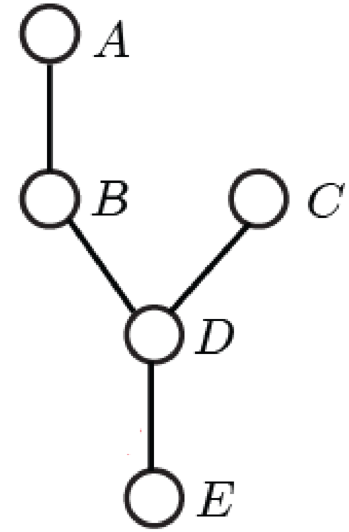
$$p(E) = \sum_A \sum_B \sum_C \sum_D p(A,B,C,D,E)$$

$$= \sum_A \sum_B \sum_C \sum_D \frac{1}{Z} f_1(A,B) \cdot f_2(B,D) \cdot f_3(C,D) \cdot f_4(D,E)$$

$$= \frac{1}{Z}\left(\sum_D f_4(D,E) \cdot \left(\sum_C f_3(C,D)\right) \cdot \left(\sum_B f_2(B,D) \cdot \left(\sum_A f_1(A,B)\right)\right)\right)$$

Marginalization with Messages

- **Use messages to express the marginalization:**



$$m_{A\to B} = \sum_A f_1(A,B) \qquad m_{C\to D} = \sum_C f_3(C,D)$$

$$m_{B\to D} = \sum_B f_2(B,D) m_{A\to B}(B)$$

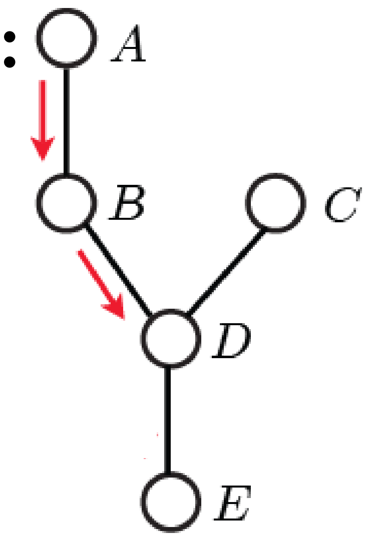
$$m_{D\to E} = \sum_D f_4(D,E) m_{B\to D}(D) m_{C\to D}(D)$$

$$p(E) = \frac{1}{Z}\left(\sum_D f_4(D,E)\cdot\left(\sum_C f_3(C,D)\right)\cdot\left(\sum_B f_2(B,D)\cdot\left(\sum_A f_1(A,B)\right)\right)\right)$$

$$= \frac{1}{Z}\left(\sum_D f_4(D,E)\cdot\left(\sum_C f_3(C,D)\right)\cdot\left(\sum_B f_2(B,D)\cdot m_{A\to B}(B)\right)\right)$$

Marginalization with Messages

- Use **messages** to express the marginalization:



$$m_{A \to B} = \sum_A f_1(A,B) \qquad m_{C \to D} = \sum_C f_3(C,D)$$

$$m_{B \to D} = \sum_B f_2(B,D) m_{A \to B}(B)$$

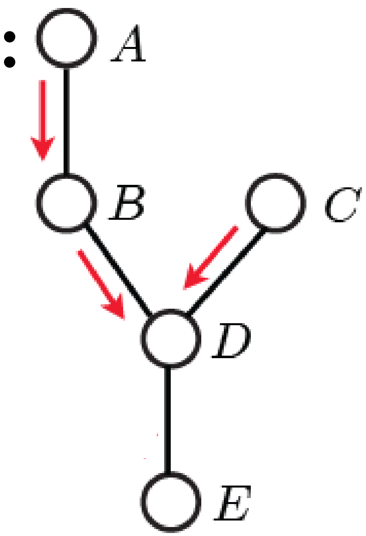
$$m_{D \to E} = \sum_D f_4(D,E) m_{B \to D}(D) m_{C \to D}(D)$$

$$p(E) = \frac{1}{Z}\left(\sum_D f_4(D,E) \cdot \left(\sum_C f_3(C,D)\right) \cdot \left(\sum_B f_2(B,D) \cdot \left(\sum_A f_1(A,B)\right)\right)\right)$$

$$= \frac{1}{Z}\left(\sum_D f_4(D,E) \cdot \left(\sum_C f_3(C,D)\right) \cdot m_{B \to D}(D)\right)$$

Marginalization with Messages

- Use **messages** to express the marginalization:



$$m_{A\to B} = \sum_A f_1(A,B) \qquad m_{C\to D} = \sum_C f_3(C,D)$$

$$m_{B\to D} = \sum_B f_2(B,D) m_{A\to B}(B)$$

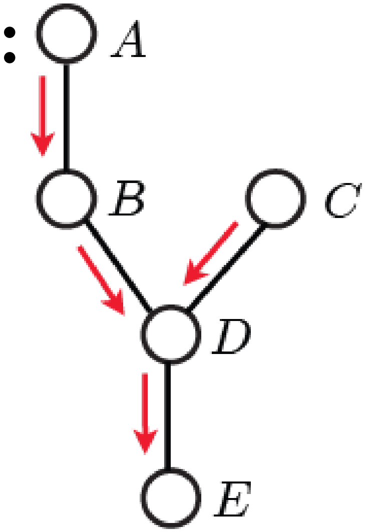
$$m_{D\to E} = \sum_D f_4(D,E) m_{B\to D}(D) m_{C\to D}(D)$$

$$p(E) = \frac{1}{Z}\left(\sum_D f_4(D,E)\cdot\left(\sum_C f_3(C,D)\right)\cdot\left(\sum_B f_2(B,D)\cdot\left(\sum_A f_1(A,B)\right)\right)\right)$$

$$= \frac{1}{Z}\left(\sum_D f_4(D,E)\cdot m_{C\to D}(D)\cdot m_{B\to D}(D)\right)$$

Marginalization with Messages

- Use **messages** to express the marginalization:



$$m_{A\to B} = \sum_A f_1(A,B) \qquad m_{C\to D} = \sum_C f_3(C,D)$$

$$m_{B\to D} = \sum_B f_2(B,D) m_{A\to B}(B)$$

$$m_{D\to E} = \sum_D f_4(D,E) m_{B\to D}(D) m_{C\to D}(D)$$

$$p(E) = \frac{1}{Z}\left(\sum_D f_4(D,E)\cdot\left(\sum_C f_3(C,D)\right)\cdot\left(\sum_B f_2(B,D)\cdot\left(\sum_A f_1(A,B)\right)\right)\right)$$

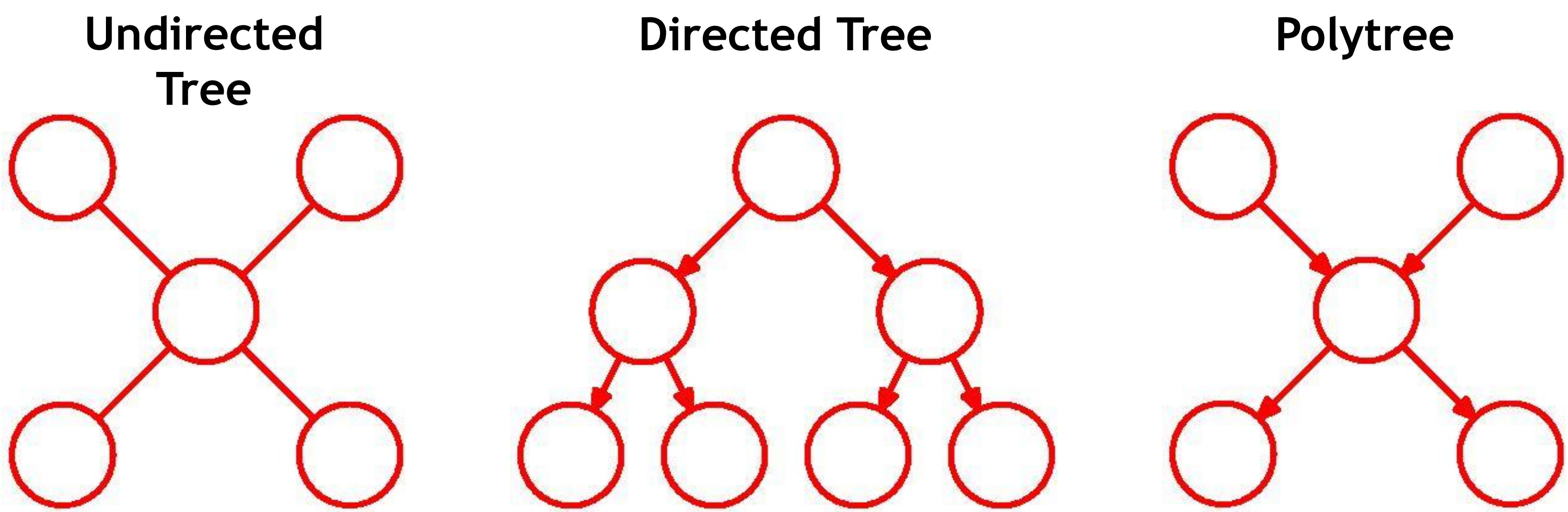
$$= \frac{1}{Z} m_{D\to E}(E)$$

Inference on Trees

- We can generalize this for all tree graphs.
  - Root the tree at the variable that we want to compute the marginal of.
  - Start computing messages at the leaves.
  - Compute the messages for all nodes for which all incoming messages have already been computed.
  - Repeat until we reach the root.

- If we want to compute the marginals for all possible nodes (roots), we can reuse some of the messages.
  - Computational expense linear in the number of nodes.

Trees – How Can We Generalize?



- Next lecture
  - Formalize the message-passing idea ⇒ Sum-product algorithm
  - Common representation of the above ⇒ Factor graphs
  - Deal with loopy graphs structures ⇒ Junction tree algorithm

Image source: C. Bishop, 2006

RWTH AACHEN UNIVERSITY

References and Further Reading

- A thorough introduction to Graphical Models in general and Bayesian Networks in particular can be found in Chapter 8 of Bishop's book.

Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006