PATTERN RECOGNITION AND MACHINE LEARNING

CHAPTER 8: GRAPHICAL MODELS

CAUSAL STRUCTURE

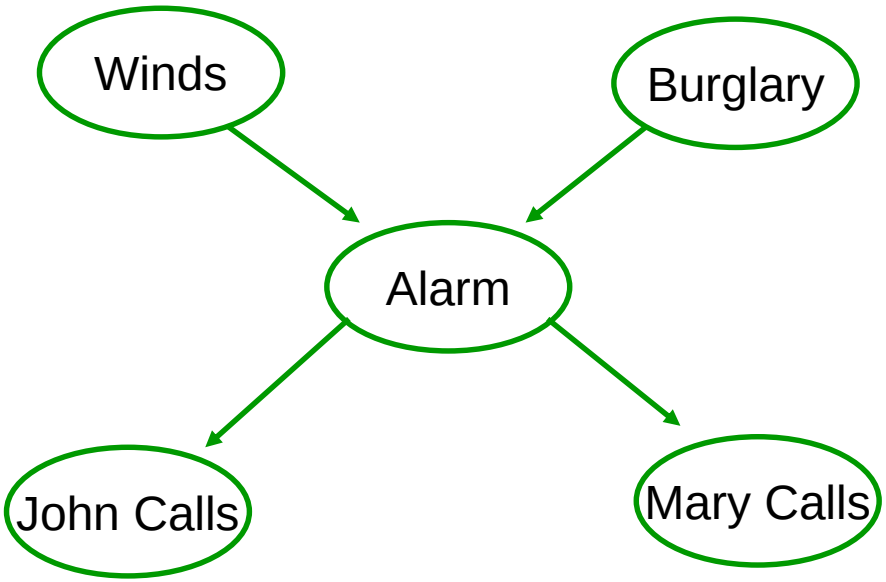
Draw a directed acyclic (causal) graph for the following

- Direct arrows from cause to effect

Story:

- There is a Burglar alarm that rings when we have Burglary
- However, sometimes it may ring because of winds that exceed 60mph
- When the alarm rings your neighbor Mary Calls
- When the alarm rings your neighbor John Calls

CAUSAL STRUCTURE



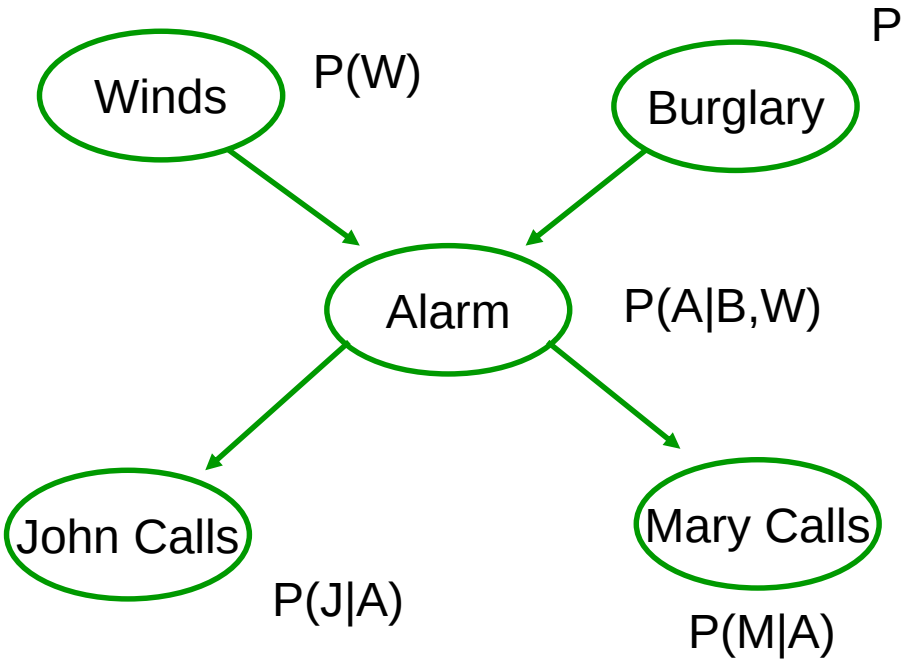
There is a Burglar alarm that rings when we have Burglary

However, sometimes it may ring because of winds that exceed 60mph

When the alarm rings your neighbor Mary Calls

When the alarm rings your neighbor John Calls

Representation of Joint Distribution

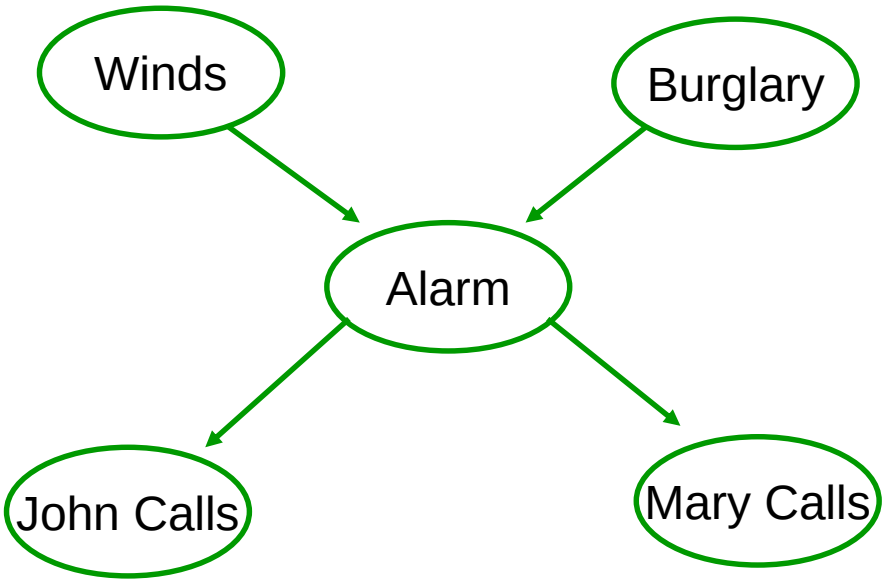


$$P(W,B,A,J,M)$$
$$=P(W)P(B)P(A|B,W)P(J|A)P(M|A)$$

In general:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \,|pa(x_i))$$

Possible Queries



Inference

P(W=?|J=True)

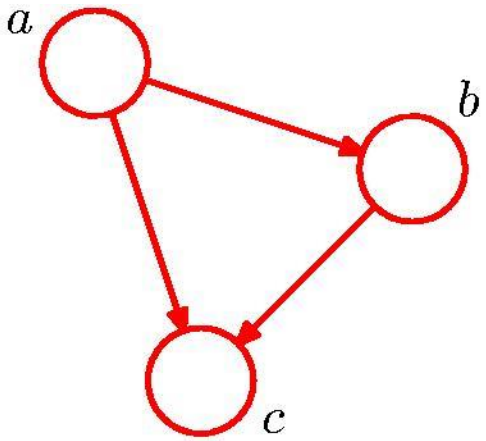
Most probable explanation

Assignment of values to all other variables that has the highest probability given that J=True and M=False

Maximum Aposteriori Hypothesis.

Bayesian Networks

Directed Acyclic Graph (DAG)

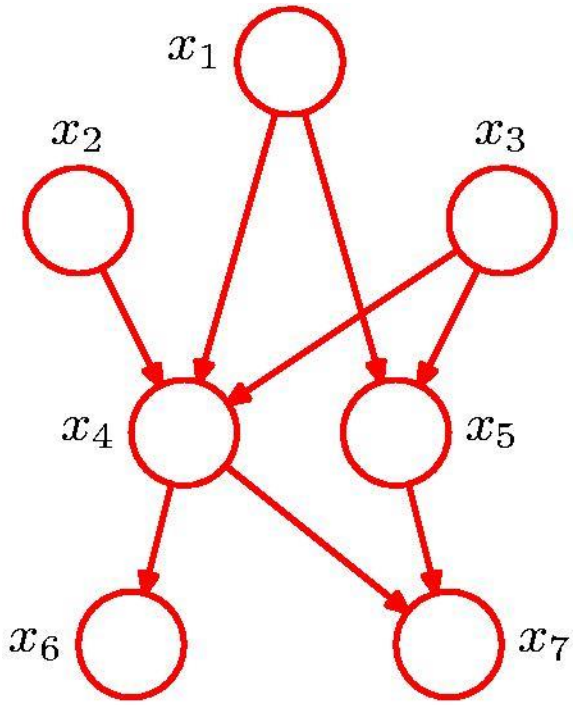


$$p(a, b, c) = p(c|a, b)p(a, b) = p(c|a, b)p(b|a)p(a)$$

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

Bayesian Networks

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



General Factorization

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

Discrete Variables (1)

General joint distribution: $K^2 - 1$ parameters

$$p(\mathbf{x}_1, \mathbf{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K}\prod_{l=1}^{K} \mu_{kl}^{x_{1k}x_{2l}}$$

Independent joint distribution: $2(K-1)$ parameters

$$\hat{p}(\mathbf{x}_1, \mathbf{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_{1k}^{x_{1k}} \prod_{l=1}^{K} \mu_{2l}^{x_{2l}}$$

Discrete Variables (2)

General joint distribution over M variables:
$K^M - 1$ parameters

M-node Markov chain: $K - 1 + (M - 1)K(K - 1)$ parameters

Conditional Independence

a is independent of b given c

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

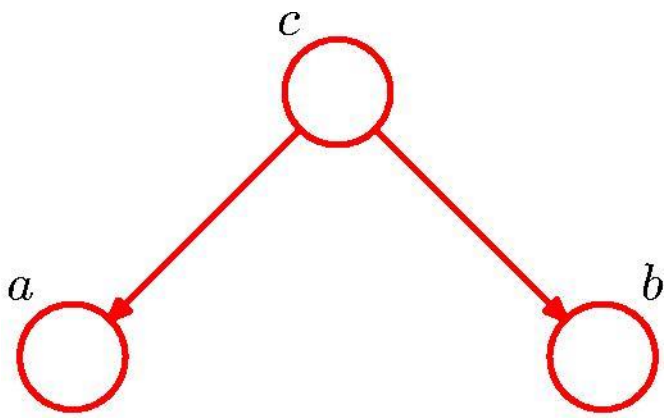
Equivalently

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

Notation

$$a \perp\!\!\!\perp b \mid c$$

Conditional Independence: Example 1

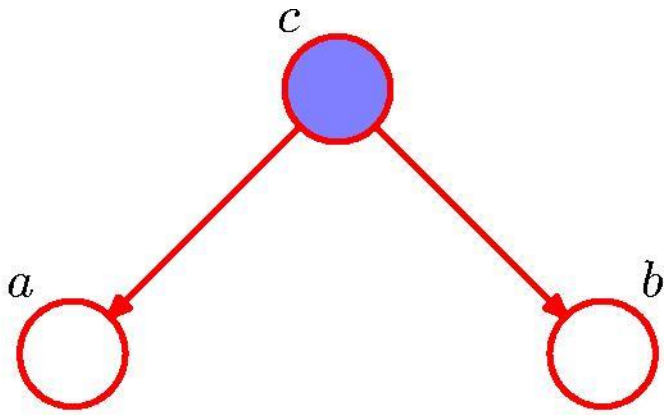


$$p(a, b, c) = p(a|c)p(b|c)p(c)$$

$$p(a, b) = \sum_c p(a|c)p(b|c)p(c)$$

$$a \not\perp\!\!\!\perp b \mid \emptyset$$

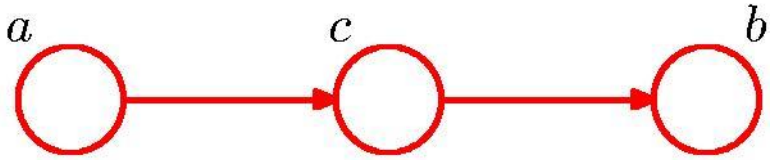
Conditional Independence: Example 1



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

$$a \perp\!\!\!\perp b \mid c$$

Conditional Independence: Example 2

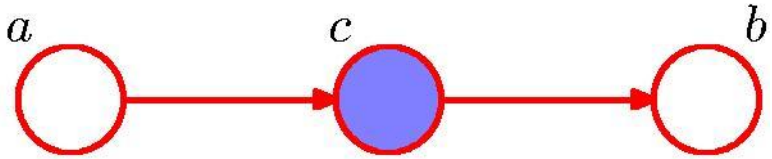


$$p(a, b, c) = p(a)p(c|a)p(b|c)$$

$$p(a, b) = p(a) \sum_c p(c|a)p(b|c) = p(a)p(b|a)$$

$$a \not\perp\!\!\!\perp b \mid \emptyset$$

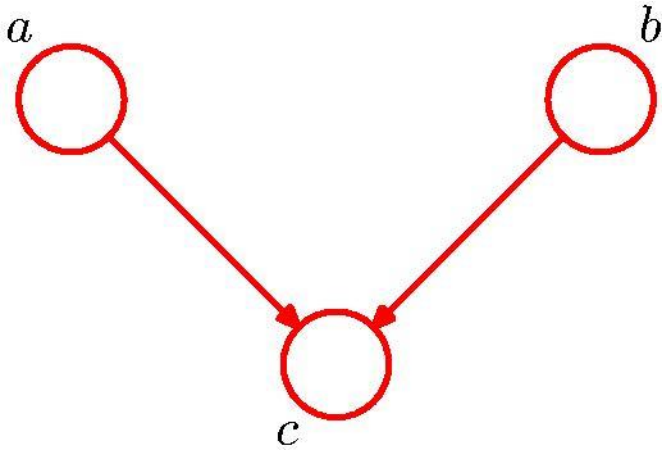
Conditional Independence: Example 2



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

$$a \perp\!\!\!\perp b \mid c$$

Conditional Independence: Example 3



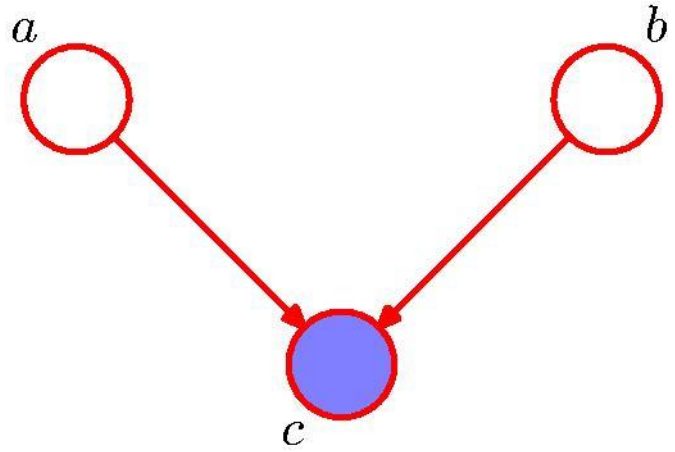
$$p(a, b, c) = p(a)p(b)p(c|a, b)$$

$$p(a, b) = p(a)p(b)$$

$$a \perp\!\!\!\perp b \mid \emptyset$$

Note: this is the opposite of Example 1, with c unobserved.

Conditional Independence: Example 3



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

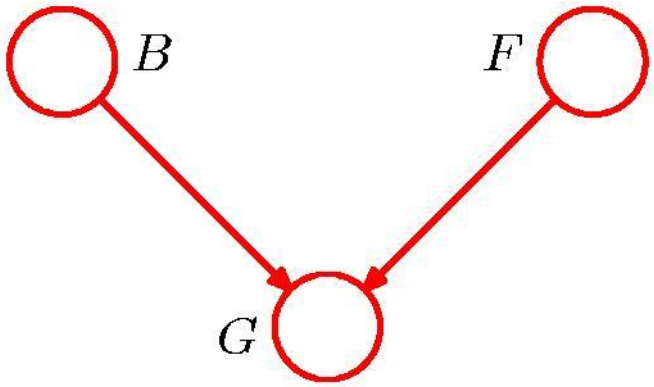
$$= \frac{p(a)p(b)p(c|a,b)}{p(c)}$$

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

Note: this is the opposite of Example 1, with c observed.

“Am I out of fuel?”

$$
\begin{aligned}
p(G=1|B=1,F=1) &= 0.8 \\
p(G=1|B=1,F=0) &= 0.2 \\
p(G=1|B=0,F=1) &= 0.2 \\
p(G=1|B=0,F=0) &= 0.1
\end{aligned}
$$



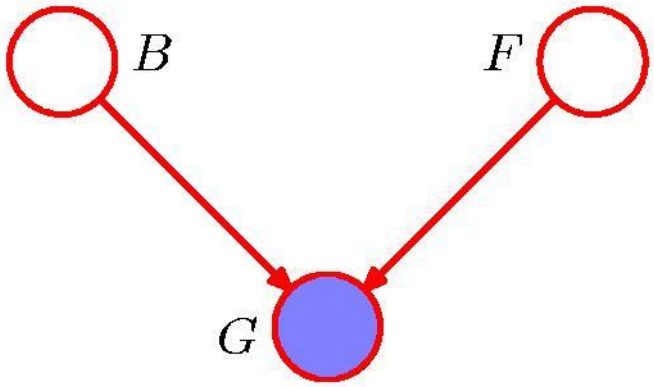
$$
\begin{aligned}
p(B=1) &= 0.9 \\
p(F=1) &= 0.9
\end{aligned}
$$

and hence

$$
p(F=0) = 0.1
$$

B = Battery (0=flat, 1=fully charged)
F = Fuel Tank (0=empty, 1=full)
G = Fuel Gauge Reading (0=empty, 1=full)

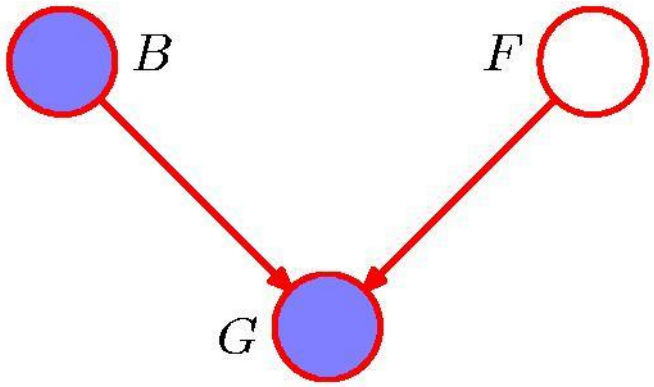
“Am I out of fuel?”



$$p(F=0|G=0) = \frac{p(G=0|F=0)p(F=0)}{p(G=0)} \simeq 0.257$$

Probability of an empty tank increased by observing $G = 0$.

“Am I out of fuel?”



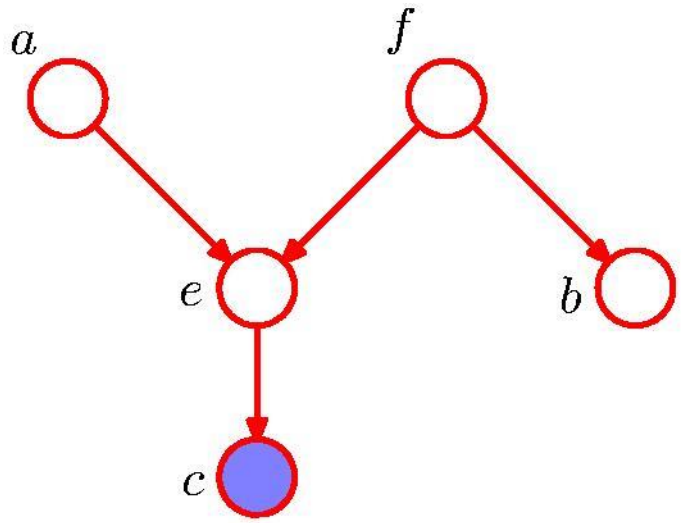
$$p(F=0|G=0,B=0) = \frac{p(G=0|B=0,F=0)p(F=0)}{\sum_{F\in\{0,1\}} p(G=0|B=0,F)p(F)}$$
$$\simeq 0.111$$

Probability of an empty tank reduced by observing $B=0$.
This referred to as “explaining away”.

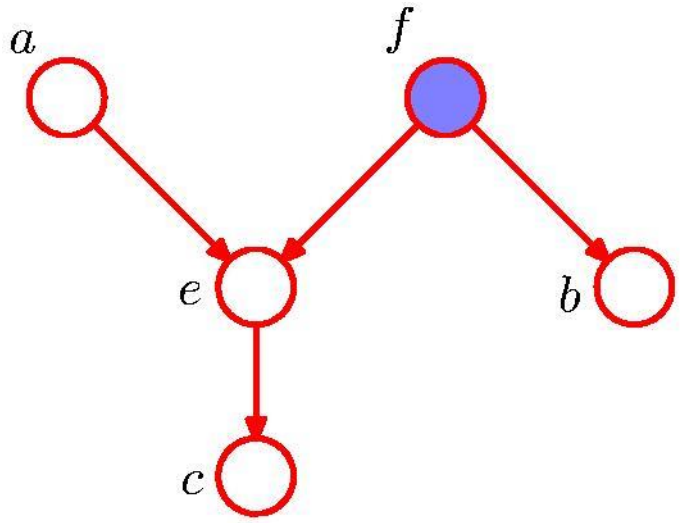
D-separation

- A, B, and C are 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
  a) 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
  b) 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 $A \perp\!\!\!\perp B \mid C$.

D-separation: Example

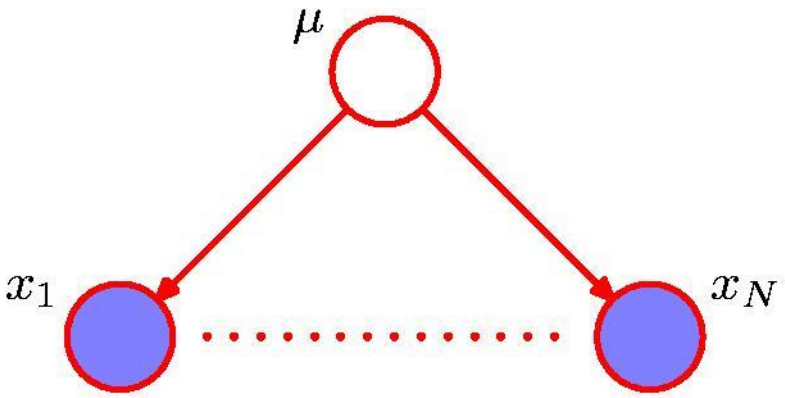


$a \not\perp\!\!\!\perp b \mid c$



$a \perp\!\!\!\perp b \mid f$

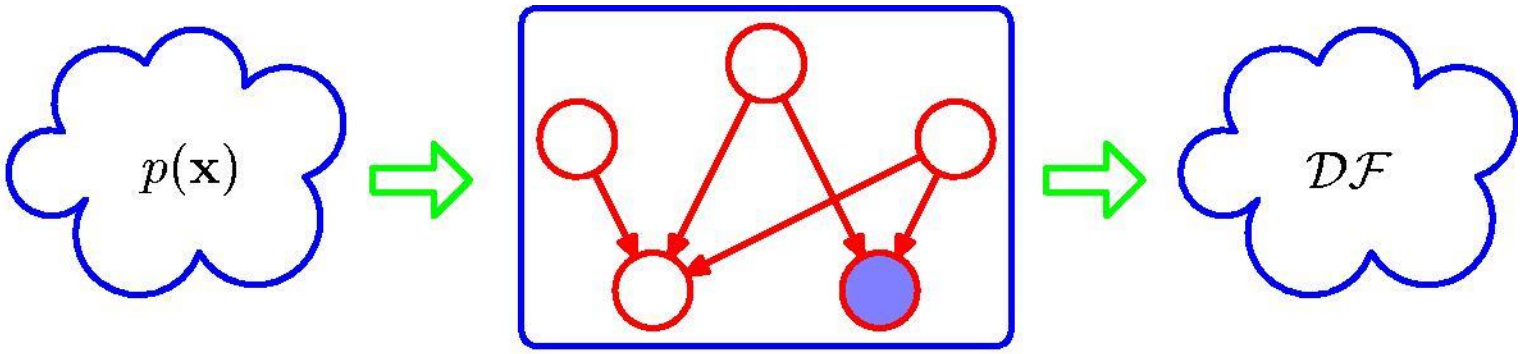
D-separation: I.I.D. Data



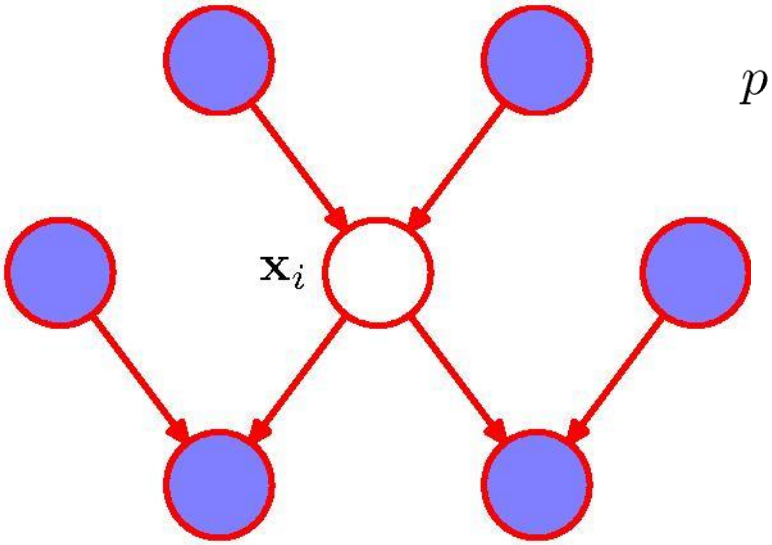
$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu)$$

$$p(\mathcal{D}) = \int_{-\infty}^{\infty} p(\mathcal{D}|\mu)p(\mu)\,\mathrm{d}\mu \neq \prod_{n=1}^{N} p(x_n)$$

Directed Graphs as Distribution Filters



The Markov Blanket

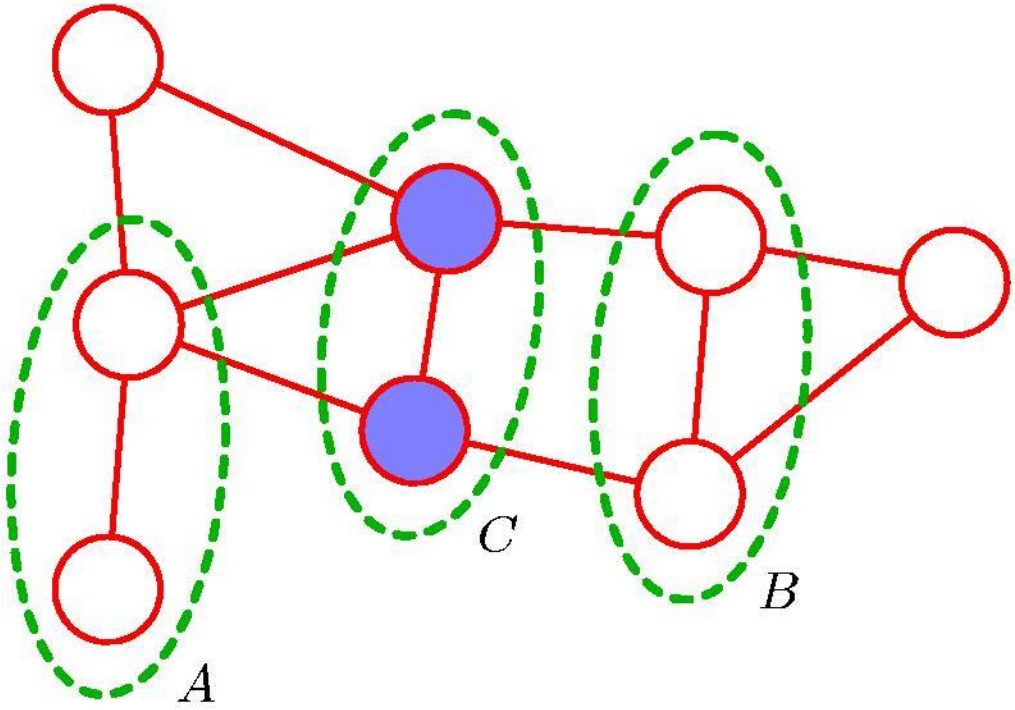


$$p(\mathbf{x}_i|\mathbf{x}_{\{j\neq i\}}) = \frac{p(\mathbf{x}_1,\ldots,\mathbf{x}_M)}{\int p(\mathbf{x}_1,\ldots,\mathbf{x}_M)\,\mathrm{d}\mathbf{x}_i}$$

$$= \frac{\prod_k p(\mathbf{x}_k|\mathrm{pa}_k)}{\int \prod_k p(\mathbf{x}_k|\mathrm{pa}_k)\,\mathrm{d}\mathbf{x}_i}$$

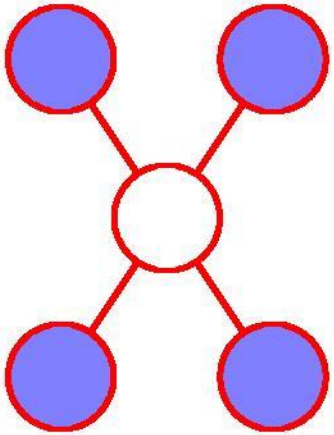
Factors independent of $\mathbf{x}_i$ cancel between numerator and denominator.

Markov Random Fields



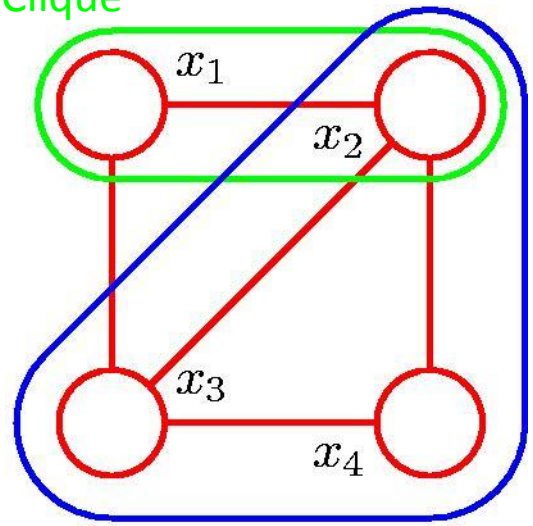
$$A \perp\!\!\!\perp B | C$$

Markov Blanket



Cliques and Maximal Cliques

Clique



Maximal Clique

Joint Distribution

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

where $\psi_C(\mathbf{x}_C)$ is the potential over clique C and

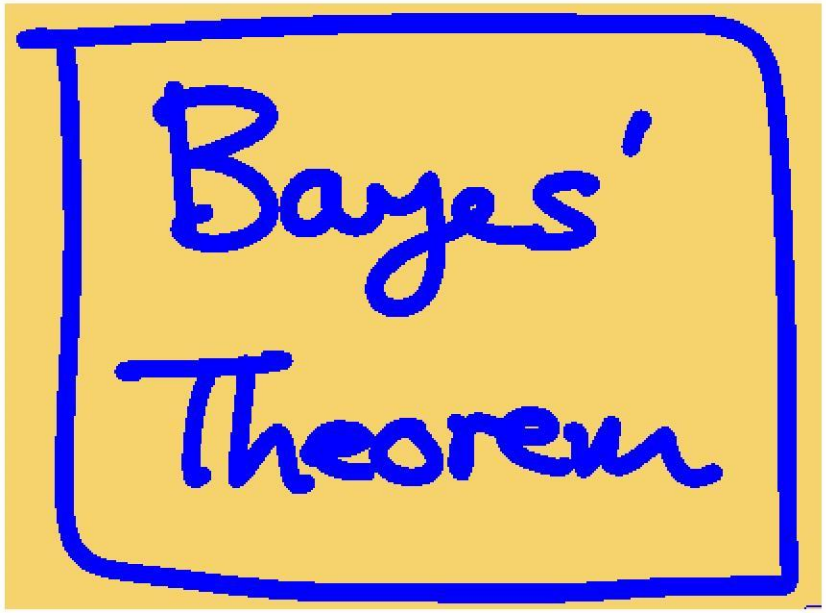
$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$$

is the normalization coefficient; note: M K-state variables $\rightarrow$ K^M terms in Z.

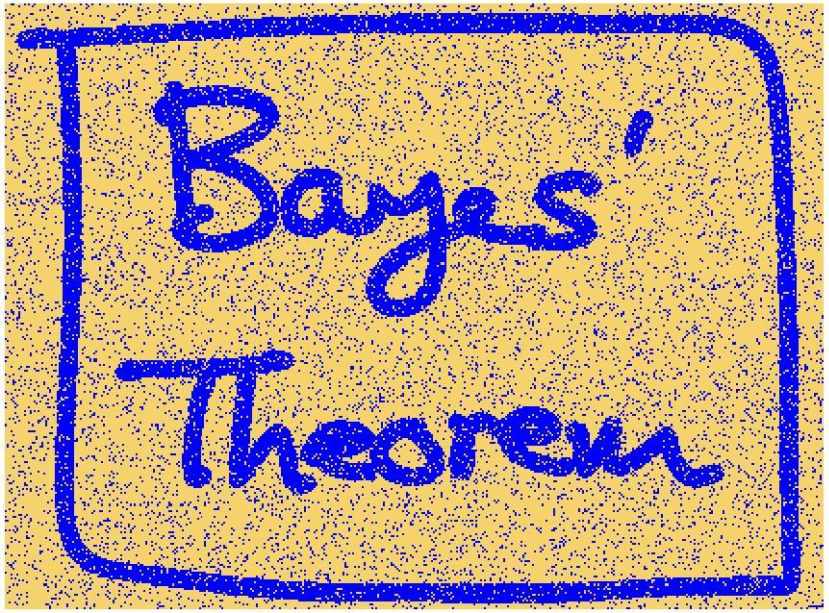
Energies and the Boltzmann distribution

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

Illustration: Image De-Noising (1)

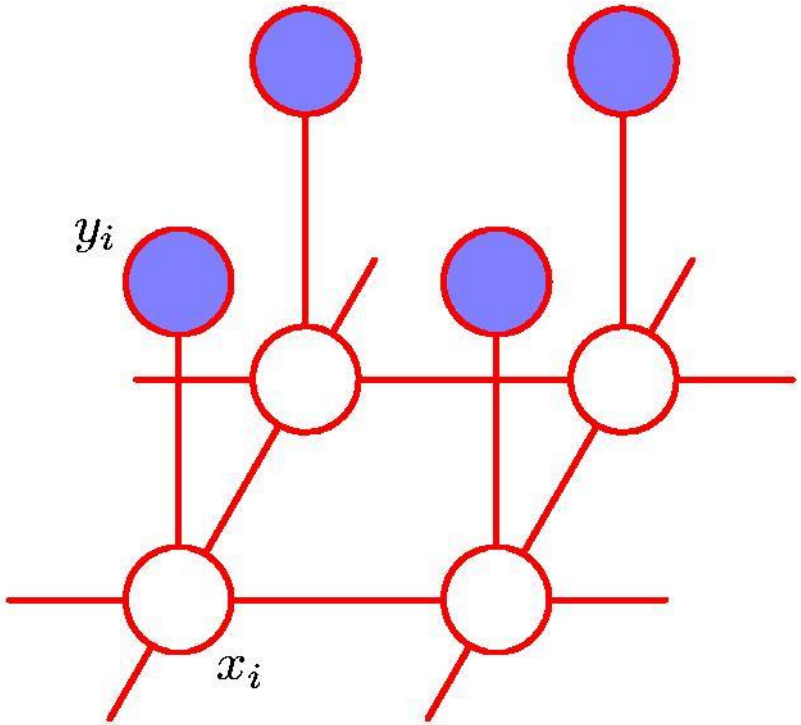


Original Image



Noisy Image

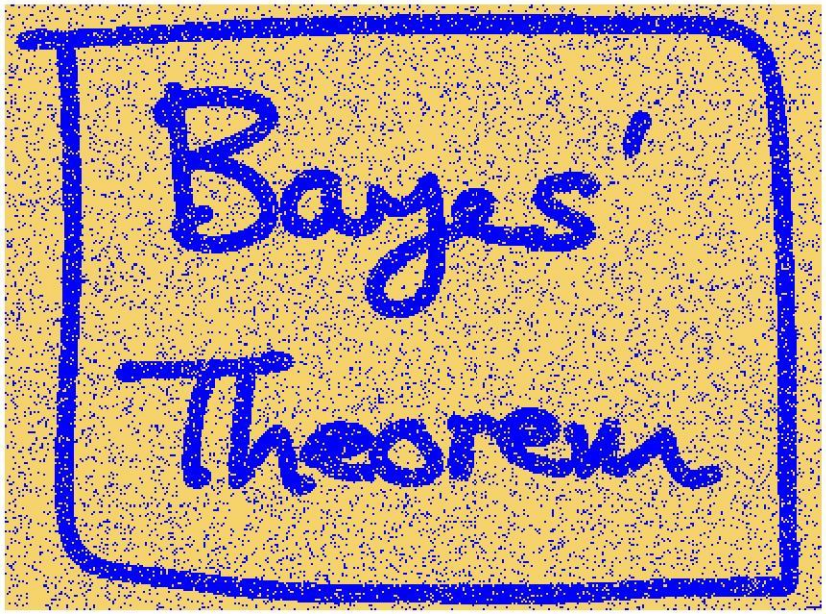
Illustration: Image De-Noising (2)



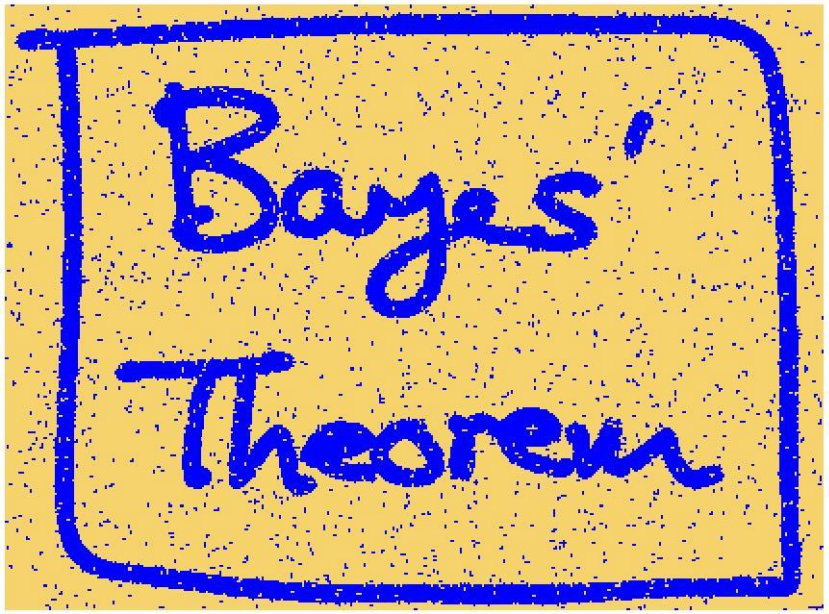
$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i$$

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

Illustration: Image De-Noising (3)

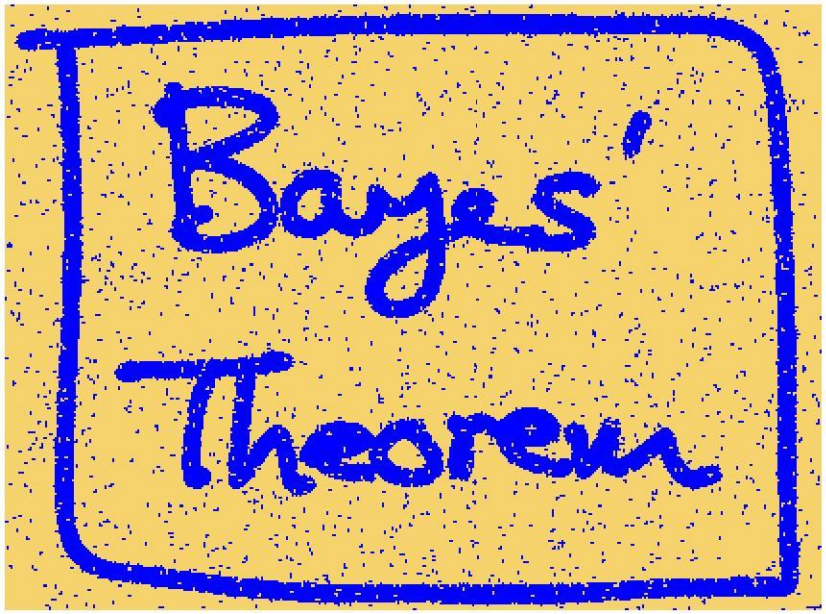


Noisy Image

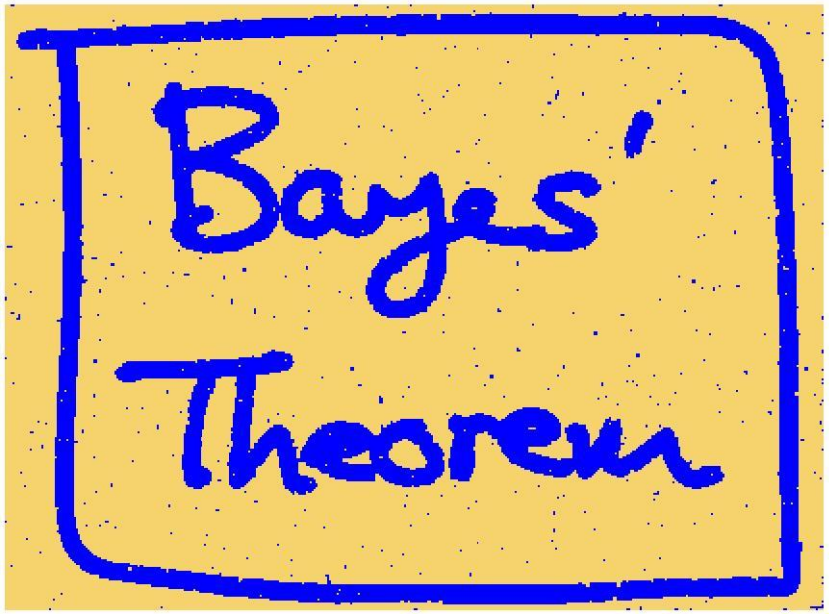


Restored Image (ICM)

Illustration: Image De-Noising (4)

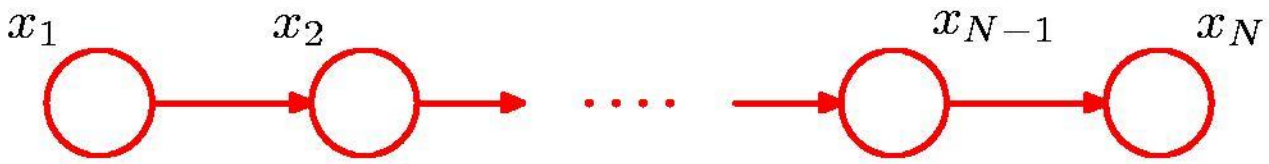


Restored Image (ICM)



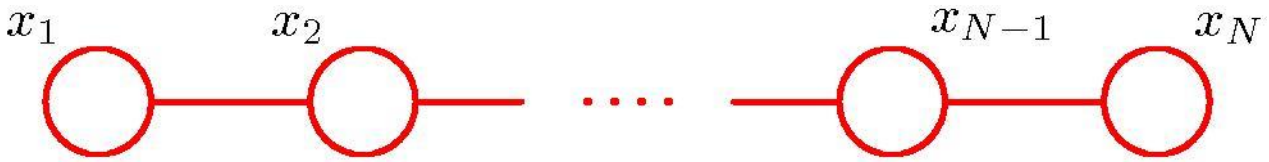
Restored Image (Graph cuts)

Converting Directed to Undirected Graphs (1)



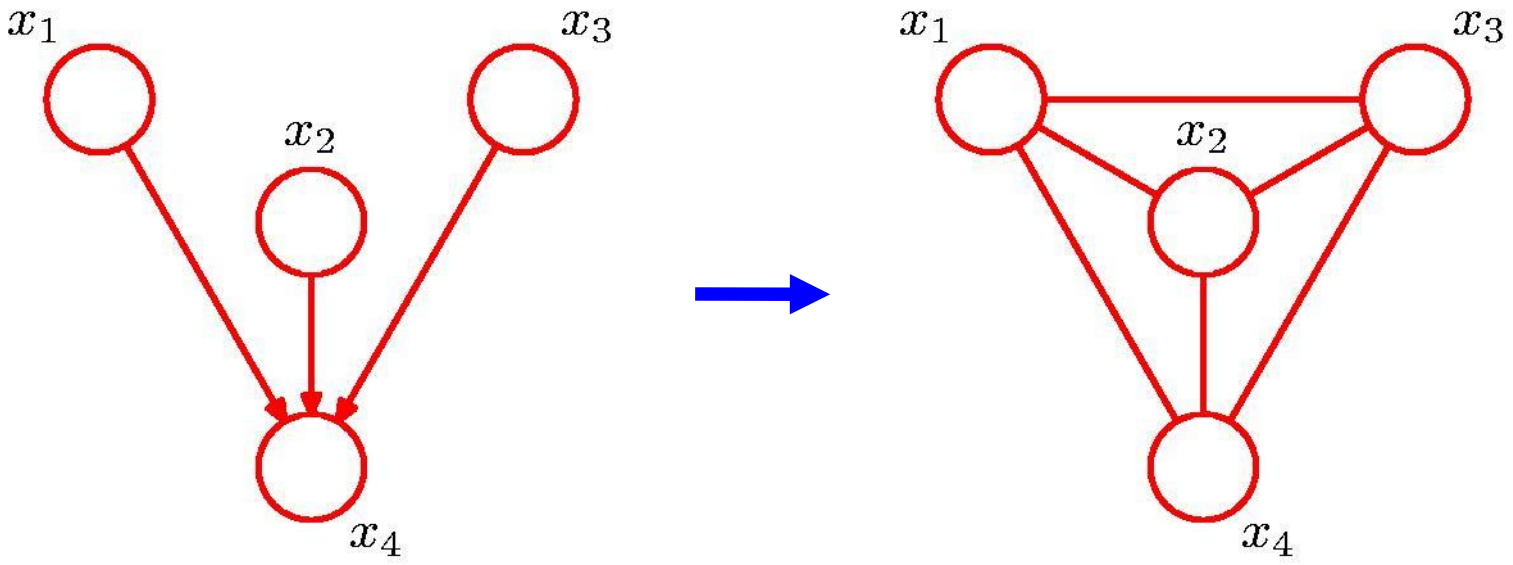
$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)\; p(x_3|x_2) \cdots p(x_N|x_{N-1})$$

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



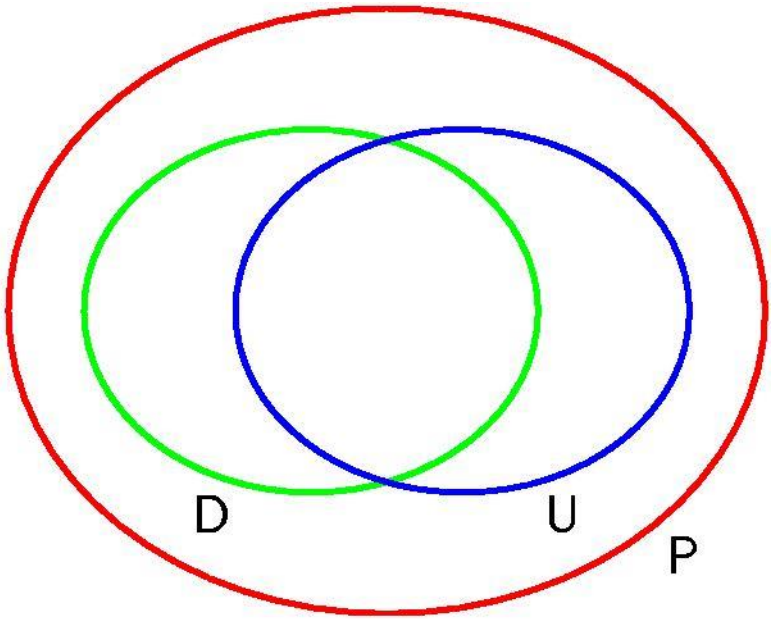
Converting Directed to Undirected Graphs (2)

Additional links

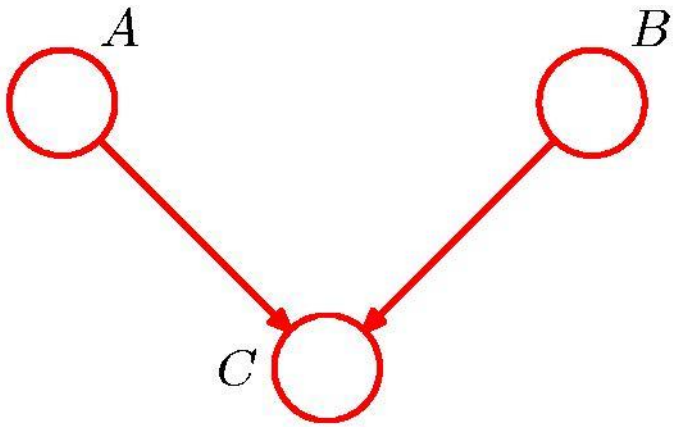


$$
\begin{aligned}
p(\mathbf{x}) &= p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3) \\
&= \frac{1}{Z}\psi_A(x_1,x_2,x_3)\psi_B(x_2,x_3,x_4)\psi_C(x_1,x_2,x_4)
\end{aligned}
$$

Directed vs. Undirected Graphs (1)

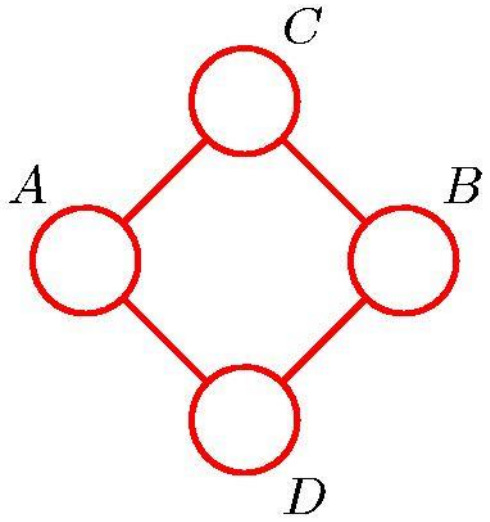


Directed vs. Undirected Graphs (2)



$$A \perp\!\!\!\perp B \mid \emptyset$$

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

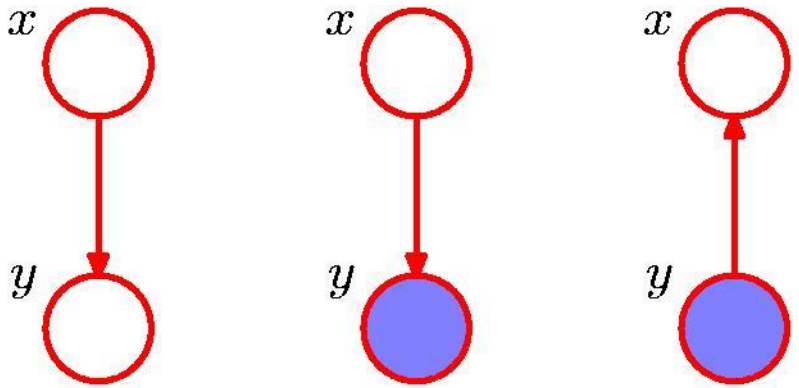


$$A \not\perp\!\!\!\perp B \mid \emptyset$$

$$A \perp\!\!\!\perp B \mid C \cup D$$

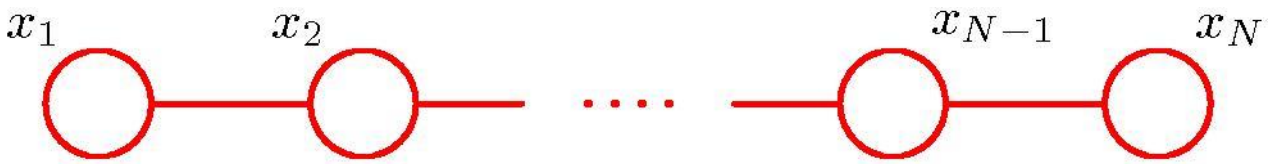
$$C \perp\!\!\!\perp D \mid A \cup B$$

Inference in Graphical Models



$$p(y) = \sum_{x'} p(y|x')p(x') \qquad p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

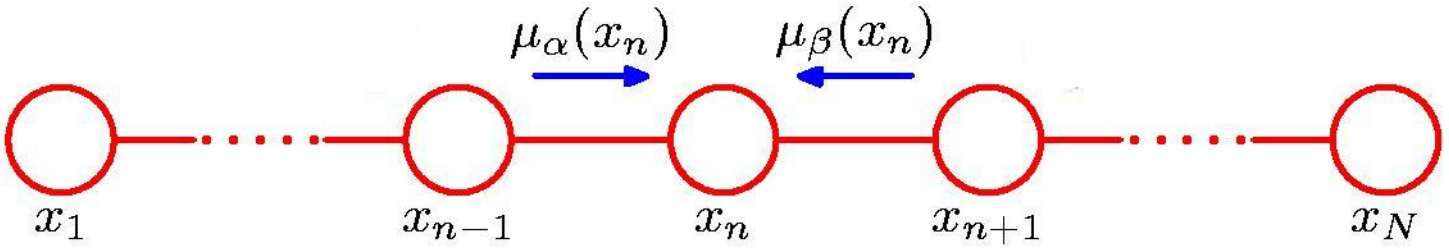
Inference on a Chain



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

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

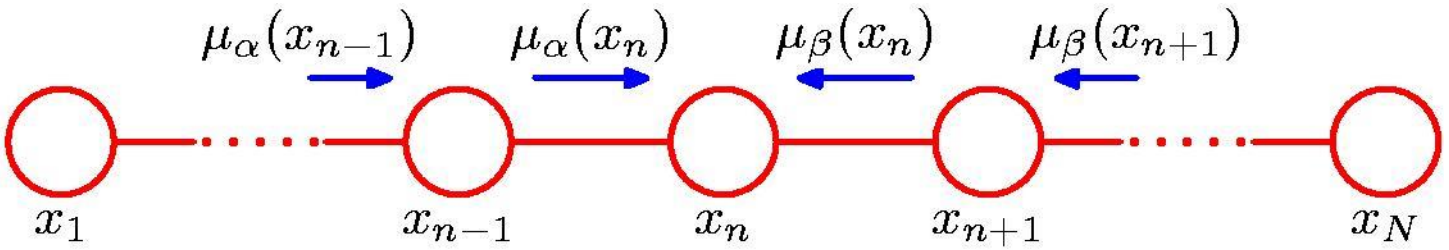
Inference on a Chain



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

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

Inference on a Chain



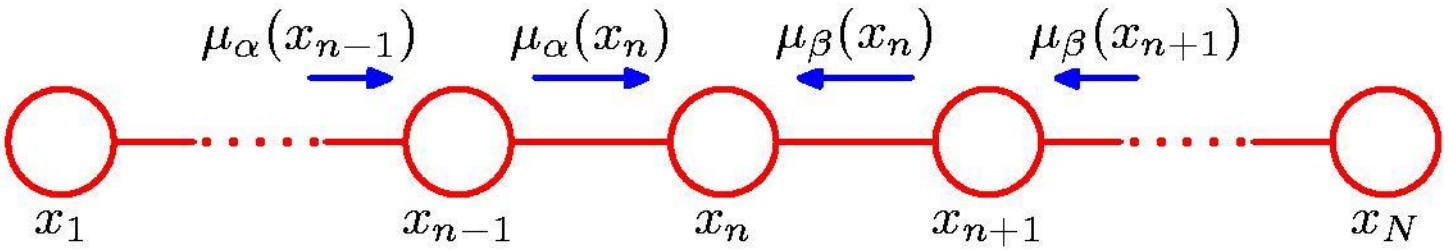
$$\mu_\alpha(x_n) = \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \left[\sum_{x_{n-2}} \cdots \right]$$

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

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

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

Inference on a Chain



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

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

Inference on a Chain

To compute local marginals:

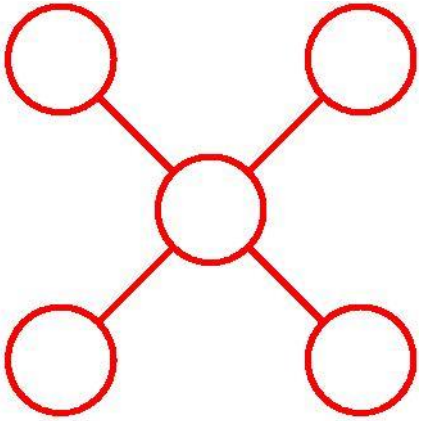
- Compute and store all forward messages, $\mu_\alpha(x_n)$.
- Compute and store all backward messages, $\mu_\beta(x_n)$.
- Compute Z at any node x_m
- Compute

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

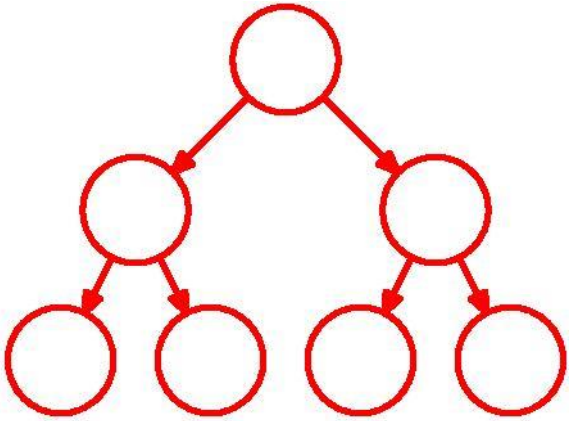
for all variables required.

Trees

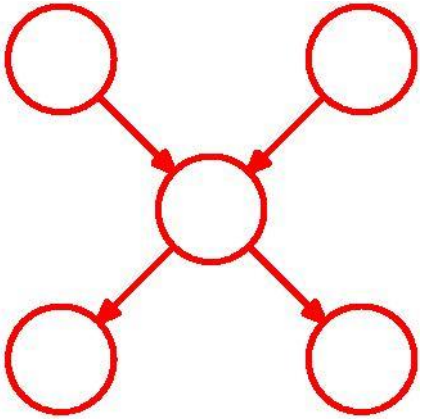
Undirected Tree



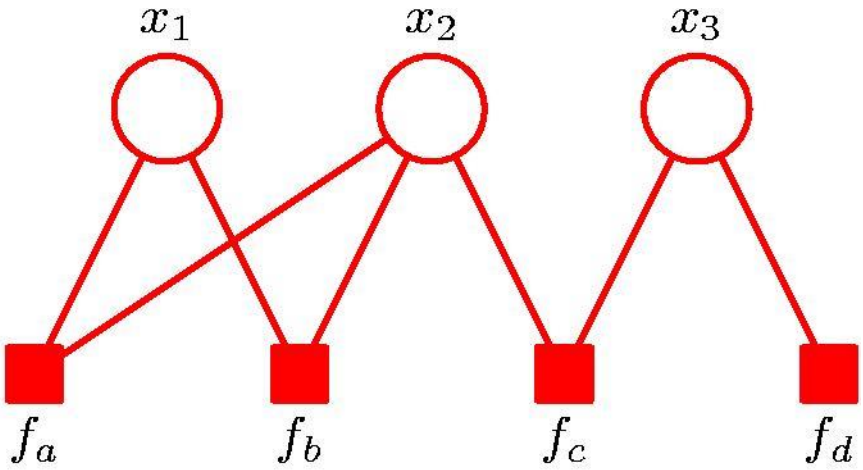
Directed Tree



Polytree



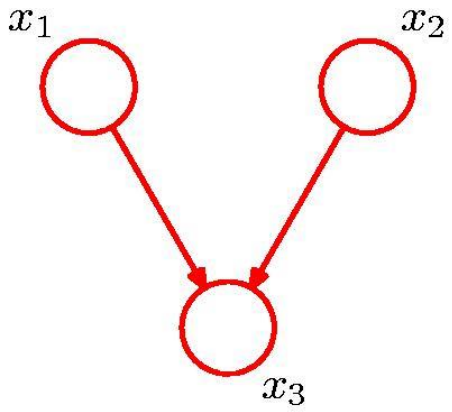
Factor Graphs

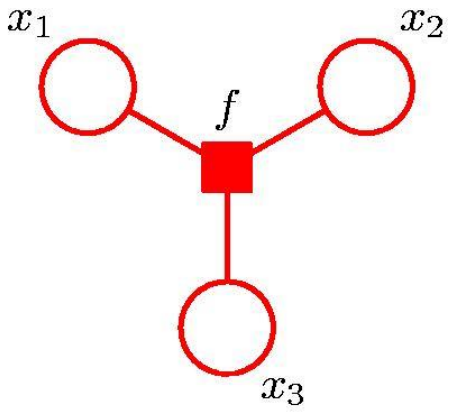


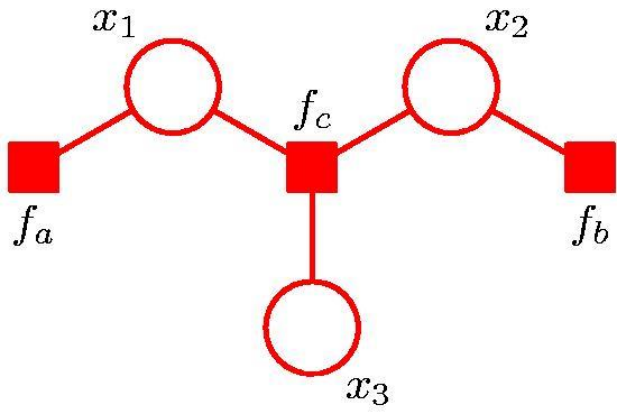
$$p(\mathbf{x}) = f_a(x_1, x_2) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3)$$

$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s)$$

Factor Graphs from Directed Graphs







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

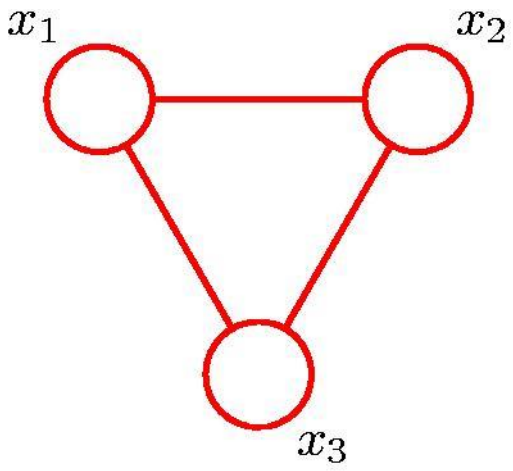
$$f(x_1,x_2,x_3) = p(x_1)p(x_2)p(_3|x_1,x_2)$$

$$f_a(x_1) = p(x_1)$$

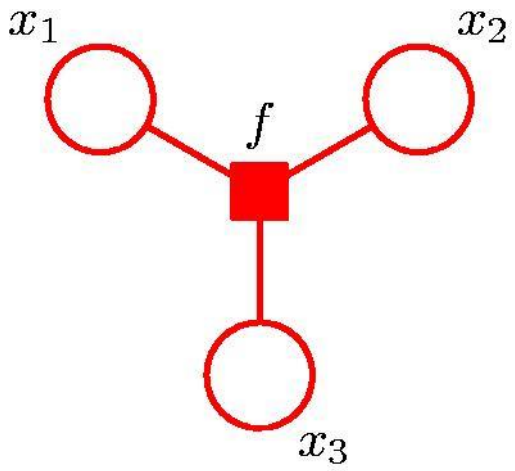
$$f_b(x_2) = p(x_2)$$

$$f_c(x_1,x_2,x_3) = p(x_3|x_1,x_2)$$

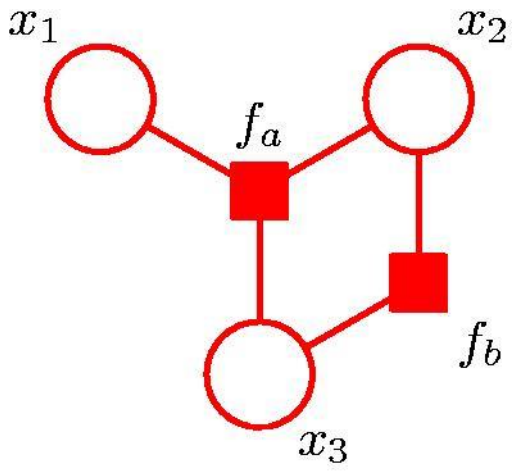
Factor Graphs from Undirected Graphs



$$\psi(x_1, x_2, x_3)$$



$$f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$$



$$f_a(x_1, x_2, x_3) f_b(x_2, x_3) = \psi(x_1, x_2, x_3)$$

The Sum-Product Algorithm (1)

Objective:

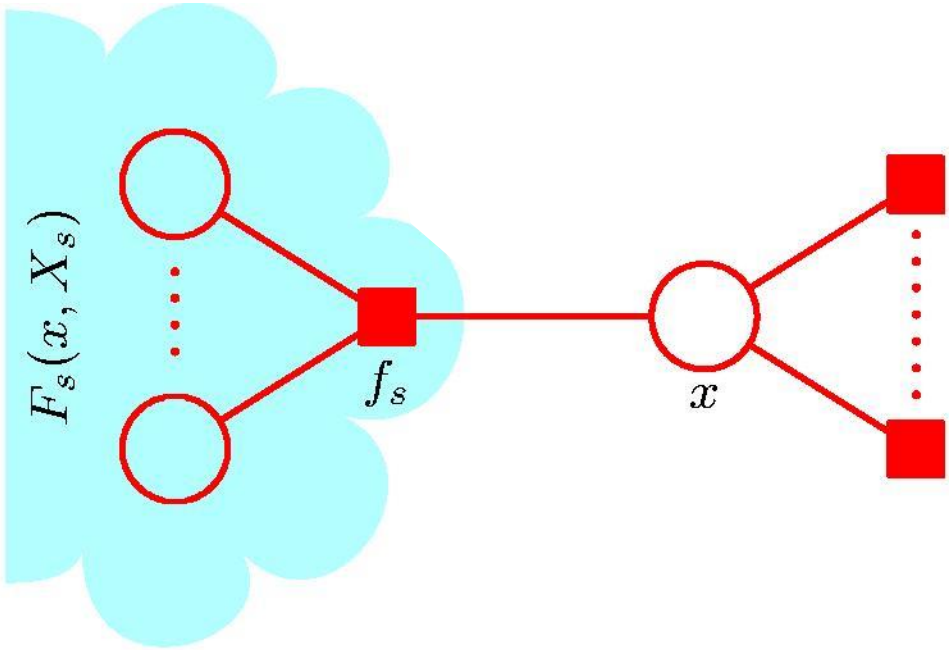
i. to obtain an efficient, exact inference algorithm for finding marginals;

ii. in situations where several marginals are required, to allow computations to be shared efficiently.

Key idea: Distributive Law

$$ab + ac = a(b + c)$$

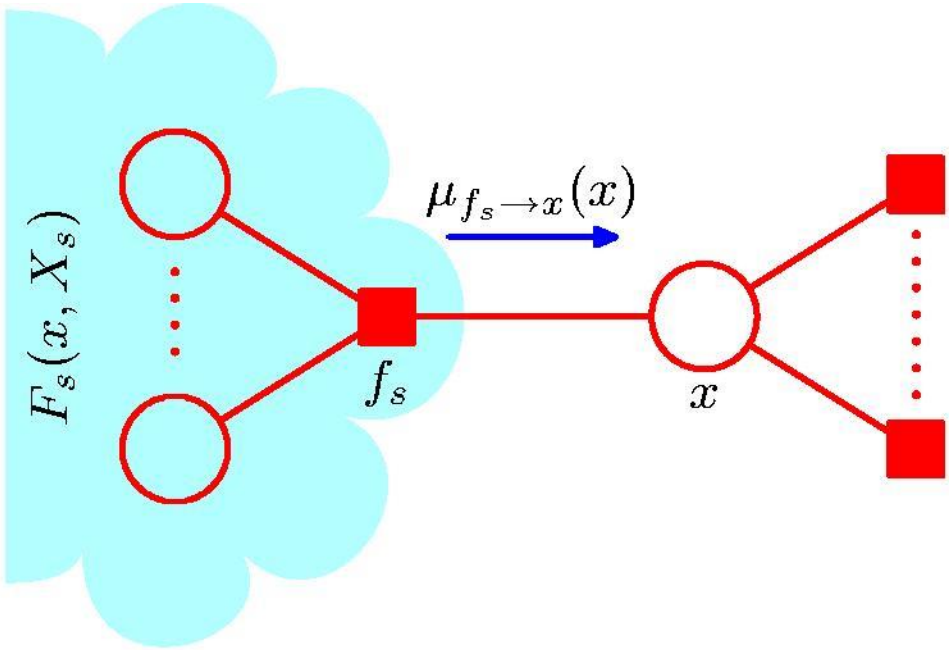
The Sum-Product Algorithm (2)



$$p(x) = \sum_{\mathbf{x}\setminus x} p(\mathbf{x})$$

$$p(\mathbf{x}) = \prod_{s\in \mathrm{ne}(x)} F_s(x, X_s)$$

The Sum-Product Algorithm (3)

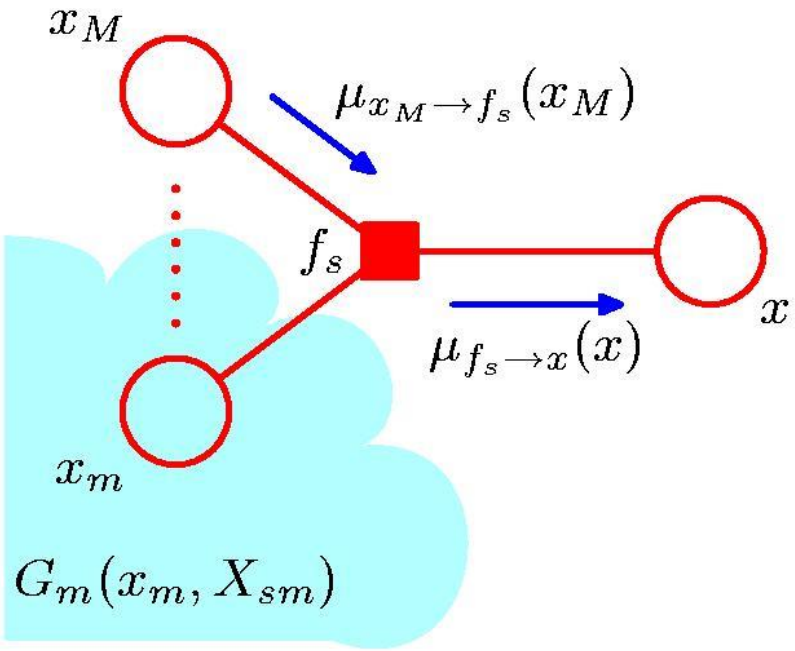


$$p(x) = \prod_{s \in \mathrm{ne}(x)} \left[\sum_{X_s} F_s(x, X_s) \right]$$

$$= \prod_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x).$$

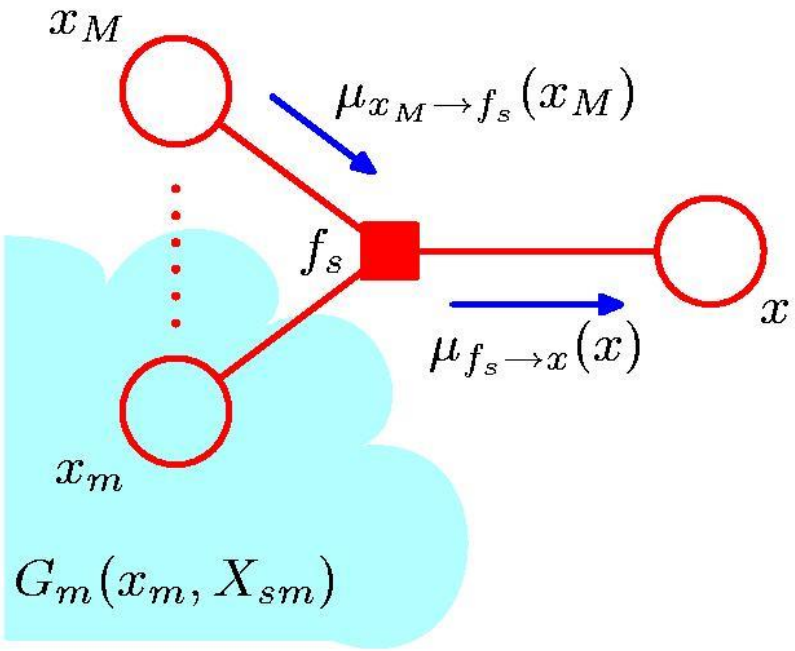
$$\mu_{f_s \to x}(x) \equiv \sum_{X_s} F_s(x, X_s)$$

The Sum-Product Algorithm (4)



$$F_s(x, X_s) = f_s(x, x_1, \ldots, x_M) G_1(x_1, X_{s1}) \ldots G_M(x_M, X_{sM})$$

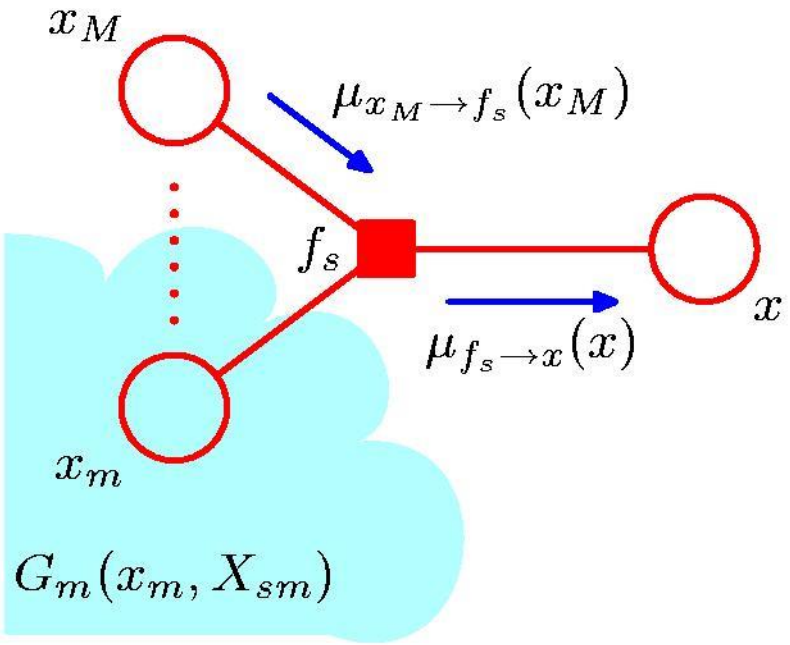
The Sum-Product Algorithm (5)



$$\mu_{f_s \to x}(x) = \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \left[\sum_{X_{sm}} G_m(x_m, X_{sm}) \right]$$

$$= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \mu_{x_m \to f_s}(x_m)$$

The Sum-Product Algorithm (6)

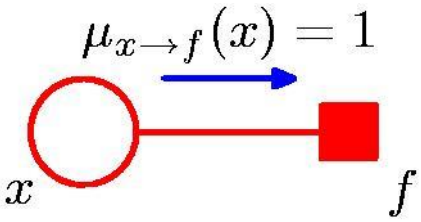


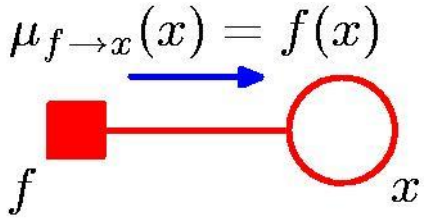
$$\mu_{x_m \to f_s}(x_m) \equiv \sum_{X_{sm}} G_m(x_m, X_{sm}) = \sum_{X_{sm}} \prod_{l \in \mathrm{ne}(x_m) \setminus f_s} F_l(x_m, X_{ml})$$

$$= \prod_{l \in \mathrm{ne}(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

The Sum-Product Algorithm (7)

Initialization



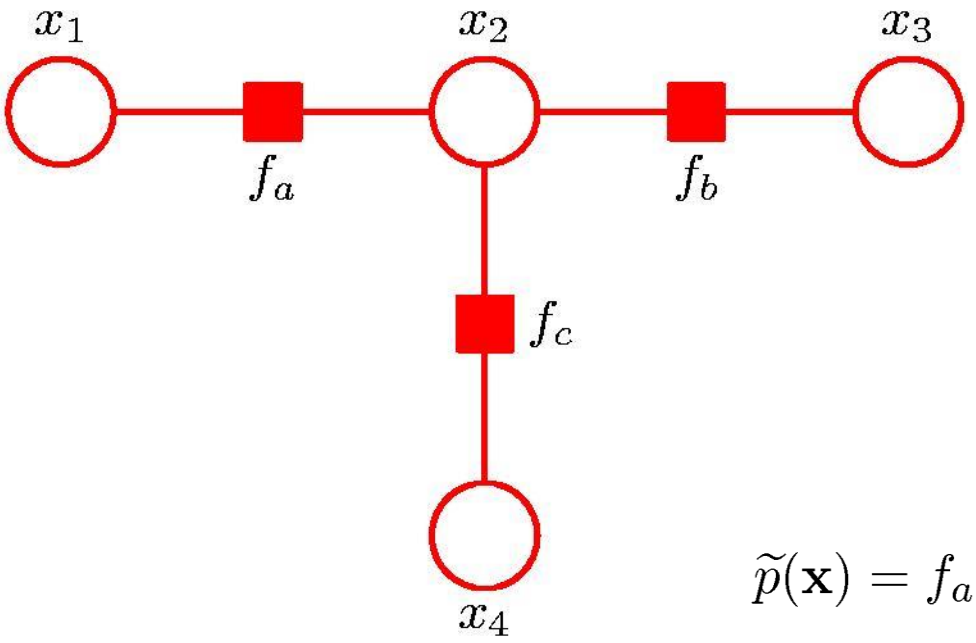


The Sum-Product Algorithm (8)

To compute local marginals:

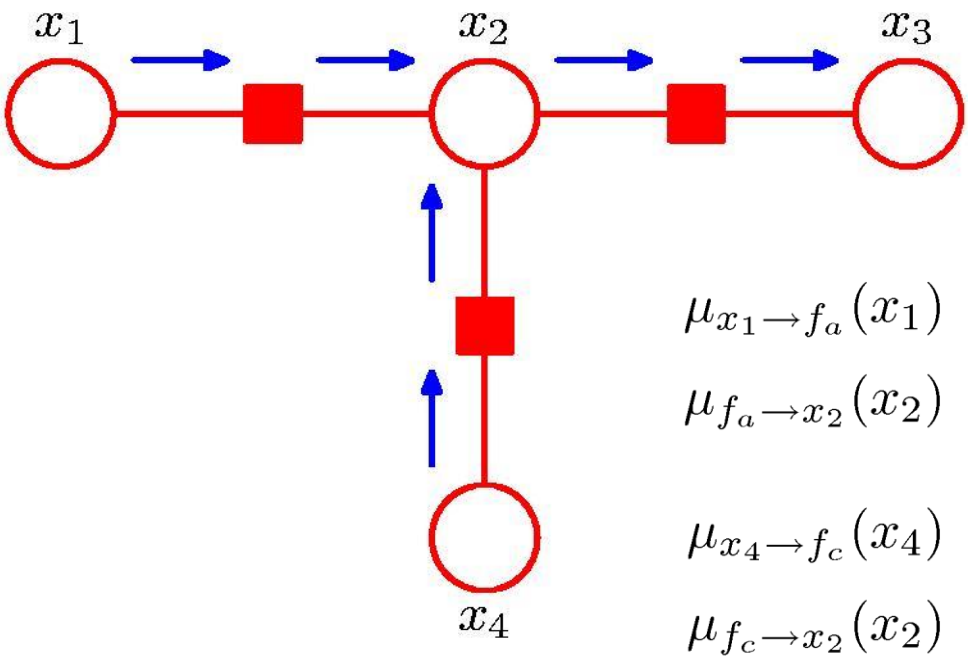
- Pick an arbitrary node as root
- Compute and propagate messages from the leaf nodes to the root, storing received messages at every node.
- Compute and propagate messages from the root to the leaf nodes, storing received messages at every node.
- Compute the product of received messages at each node for which the marginal is required, and normalize if necessary.

Sum-Product: Example (1)



$$\widetilde{p}(\mathbf{x}) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4)$$

Sum-Product: Example (2)



$$\mu_{x_1 \to f_a}(x_1) = 1$$

$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2)$$

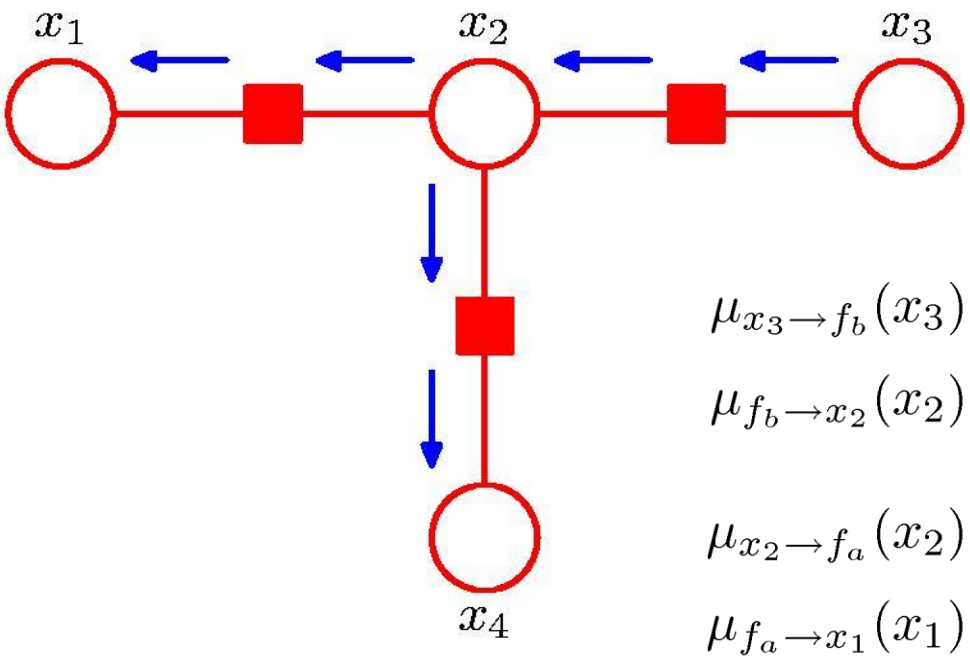
$$\mu_{x_4 \to f_c}(x_4) = 1$$

$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4)$$

$$\mu_{x_2 \to f_b}(x_2) = \mu_{f_a \to x_2}(x_2)\mu_{f_c \to x_2}(x_2)$$

$$\mu_{f_b \to x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3)\mu_{x_2 \to f_b}(x_2)$$

Sum-Product: Example (3)



$$\mu_{x_3 \to f_b}(x_3) = 1$$

$$\mu_{f_b \to x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3)$$

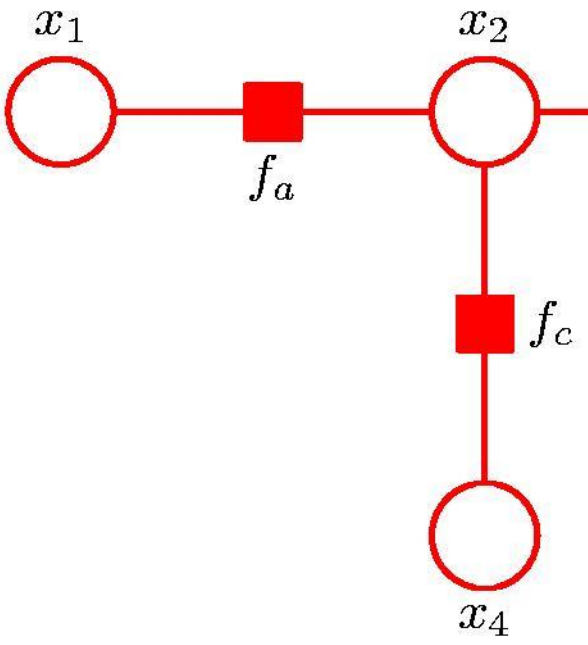
$$\mu_{x_2 \to f_a}(x_2) = \mu_{f_b \to x_2}(x_2)\mu_{f_c \to x_2}(x_2)$$

$$\mu_{f_a \to x_1}(x_1) = \sum_{x_2} f_a(x_1, x_2)\mu_{x_2 \to f_a}(x_2)$$

$$\mu_{x_2 \to f_c}(x_2) = \mu_{f_a \to x_2}(x_2)\mu_{f_b \to x_2}(x_2)$$

$$\mu_{f_c \to x_4}(x_4) = \sum_{x_2} f_c(x_2, x_4)\mu_{x_2 \to f_c}(x_2)$$

Sum-Product: Example (4)



$$\begin{aligned}
\widetilde{p}(x_2) &= \mu_{f_a \to x_2}(x_2)\mu_{f_b \to x_2}(x_2)\mu_{f_c \to x_2}(x_2) \\
&= \left[\sum_{x_1} f_a(x_1, x_2)\right] \left[\sum_{x_3} f_b(x_2, x_3)\right] \\
&\quad \left[\sum_{x_4} f_c(x_2, x_4)\right] \\
&= \sum_{x_1}\sum_{x_3}\sum_{x_4} f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4) \\
&= \sum_{x_1}\sum_{x_3}\sum_{x_4} \widetilde{p}(\mathbf{x})
\end{aligned}$$

The Max-Sum Algorithm

Objective: an efficient algorithm for finding

i. the value $\mathbf{x}^{\max}$ that maximises $p(\mathbf{x})$;

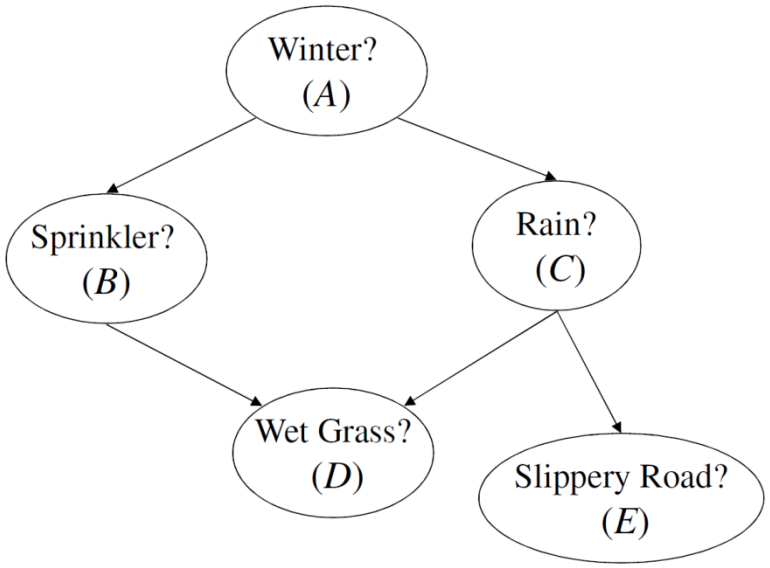
ii. the value of $p(\mathbf{x}^{\max})$.

In general, maximum marginals ≠ joint maximum.

	$x = 0$	$x = 1$
$y = 0$	0.3	0.4
$y = 1$	0.3	0.0

$$\arg\max_x p(x, y) = 1 \qquad \arg\max_x p(x) = 0$$

Possible Queries



- Probability of Evidence
  - P(E=True)=?
- Marginal Estimation
  - P(A=?|E=True)=?
- Most probable explanation
  - Assignment of values to all other variables that has the highest probability given that A=True and E=False
- Maximum Aposteriori Hypothesis.

Inference Algorithms: Algorithms that take a Bayesian network as input and output an answer to the query.

Inference Algorithms

Exact Algorithm

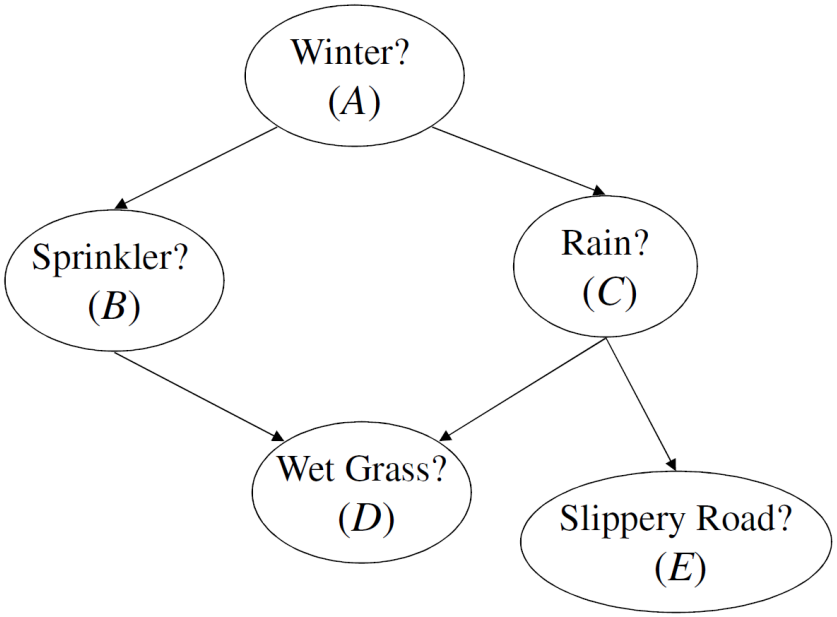
- **Variable Elimination**

Approximate Algorithms

- Belief Propagation
- Importance sampling
- Markov Chain Monte Carlo sampling

Variable Elimination

- One of the simplest algorithms for inference in Bayesian networks
  - Successively remove variables from the Bayesian network until only the query variables remain



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

Joint Probability Distribution

A	*B*	*C*	*D*	*E*	Pr(.)
true	true	true	true	true	0.06384
true	true	true	true	false	0.02736
true	true	true	false	true	0.00336
true	true	true	false	false	0.00144
true	true	false	true	true	0.0
true	true	false	true	false	0.02160
true	true	false	false	true	0.0
true	true	false	false	false	0.00240
true	false	true	true	true	0.21504
true	false	true	true	false	0.09216
true	false	true	false	true	0.05376
true	false	true	false	false	0.02304
true	false	false	true	true	0.0
true	false	false	true	false	0.0
true	false	false	false	true	0.0
true	false	false	false	false	0.09600
false	true	true	true	true	0.01995
false	true	true	true	false	0.00855
false	true	true	false	true	0.00105
false	true	true	false	false	0.00045
false	true	false	true	true	0.0
false	true	false	true	false	0.24300
false	true	false	false	true	0.0
false	true	false	false	false	0.02700
false	false	true	true	true	0.00560
false	false	true	true	false	0.00240
false	false	true	false	true	0.00140
false	false	true	false	false	0.00060
false	false	false	true	true	0.0
false	false	false	true	false	0.0
false	false	false	false	true	0.0
false	false	false	false	false	0.0900

P(D=true,E=true)=?
P(A=true|D=true,E=true)
=?

How does the algorithm work?
Task: Computing probability of evidence

Instantiate Evidence variables and remove them from all conditional probability tables

Select an ordering of variables

Eliminate variables one by one along the ordering

How to eliminate a variable ?

Multiply all functions/factors that mention the variable yielding a function f

Sum-out the variable from f yielding a function f'

Add f' to the set of original functions

Multiplication of factors

A	B	C	ϕ(A,B,C)
0	0	0	3
0	0	1	2
0	1	0	1
0	1	1	5
1	0	0	3
1	0	1	8
1	1	0	6
1	1	1	3

$\times$

A	C	D	ϕ(A,C,D)
0	0	0	4
0	0	1	2
0	1	0	11
0	1	1	4
1	0	0	2
1	0	1	1
1	1	0	5
1	1	1	1

$=$

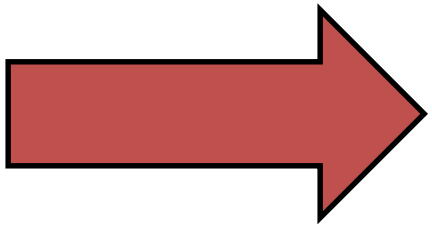
A	B	C	D	ϕ(A,B,C,D)
0	0	0	0	3*4=12
0	0	0	1	3*2=6
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Complexity is the size of the product table (exp (w)) times the number of factors (m) where w is the cardinality of the union of the scopes of functions

Summing out a set of variables

A	B	C	ϕ(A,B,C)
0	0	0	3
0	0	1	2
0	1	0	1
0	1	1	5
1	0	0	3
1	0	1	8
1	1	0	6
1	1	1	3

Sum-out B and C



A	ϕ(A)
0	
1	

$$\sum_{b,c} \phi(a,b,c)$$

Complexity is the size of the table : exp (w)

The Formal Algorithm

input:

$\mathcal{N}$: Bayesian network

$\mathbf{Q}$: variables in network $\mathcal{N}$

π: ordering of network variables not in $\mathbf{Q}$

1: $\mathcal{S} \leftarrow$ CPTs of network $\mathcal{N}$
2: **for** $i = 1$ to length of order π **do**
3: $\quad f \leftarrow \prod_k f_k$, where f_k belongs to $\mathcal{S}$ and mentions variable $\pi(i)$
4: $\quad f_i \leftarrow \sum_{\pi(i)} f$
5: $\quad$ replace all factors f_k in $\mathcal{S}$ by factor f_i
6: **end for**
7: **return** $\prod_{f \in \mathcal{S}} f$

Variable Elimination: Example

Compute P(D=true ,E=true)?

On the board.

A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

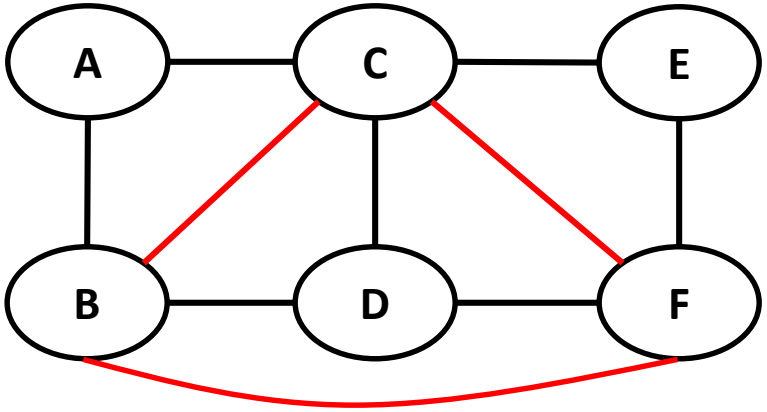
A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

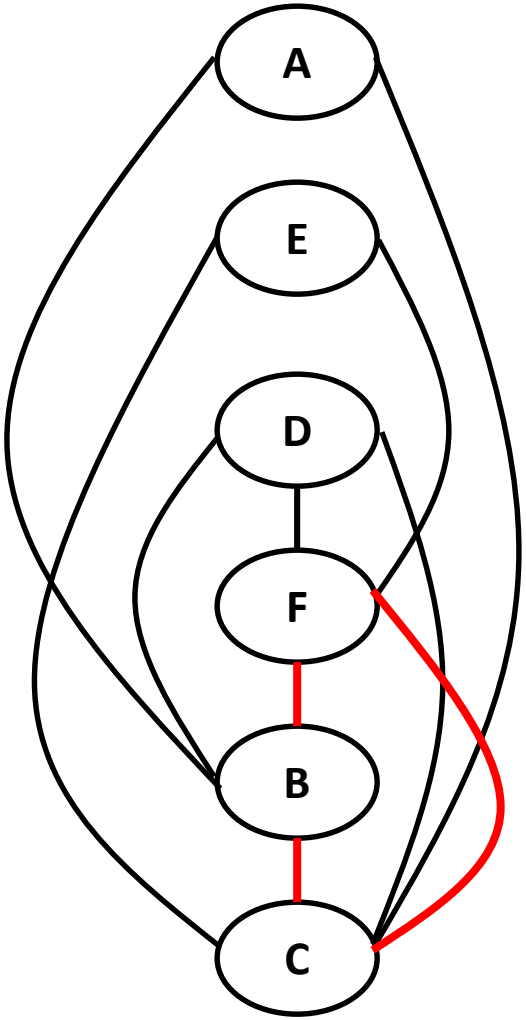
Variable Elimination: Complexity

Schematic operation on a graph

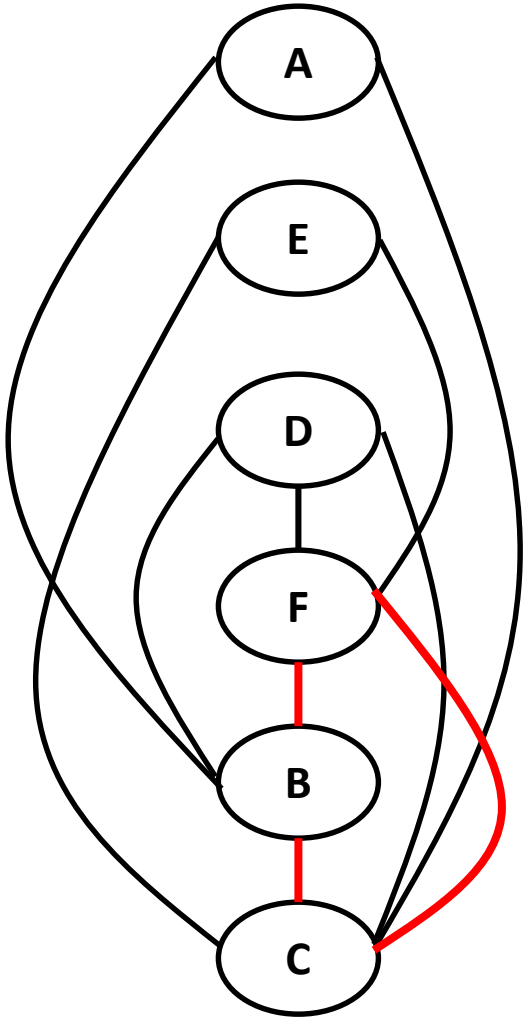


Process nodes in order

Eliminate = Connect all children of a node to each other



Variable elimination: Complexity



- Complexity of eliminating variable "i"
  - $\exp(\text{children}_i)$
- Complexity of variable elimination:
  - $\text{nexp}(\max(\text{children}_i))$
- Treewidth
  - Minimum over all possible graphs constructed this way

Variable Elimination for MPE and MAR

MARGINAL TASK

Ratio of two evidence probabilities

$P(A=a|B=b)=P(A=a,B=b)/P(B=b)$

Use VE to compute numerator and denominator

MPE TASK

Replace sum-out operation by max-out operation

MAX_S

S	C	Value
male	yes	0.05
male	no	0.95
female	yes	0.01
female	no	0.99

=

C	Value
yes	0.05
no	0.99

The Junction Tree Algorithm

- *Exact* inference on general graphs.
- Works by turning the initial graph into a *junction tree* and then running a sum-product-like algorithm.
- *Intractable* on graphs with large cliques.

Loopy Belief Propagation

- Sum-Product on general graphs.
- Initial unit messages passed across all links, after which messages are passed around until convergence (not guaranteed!).
- *Approximate* but *tractable* for large graphs.
- Sometime works well, sometimes not at all.