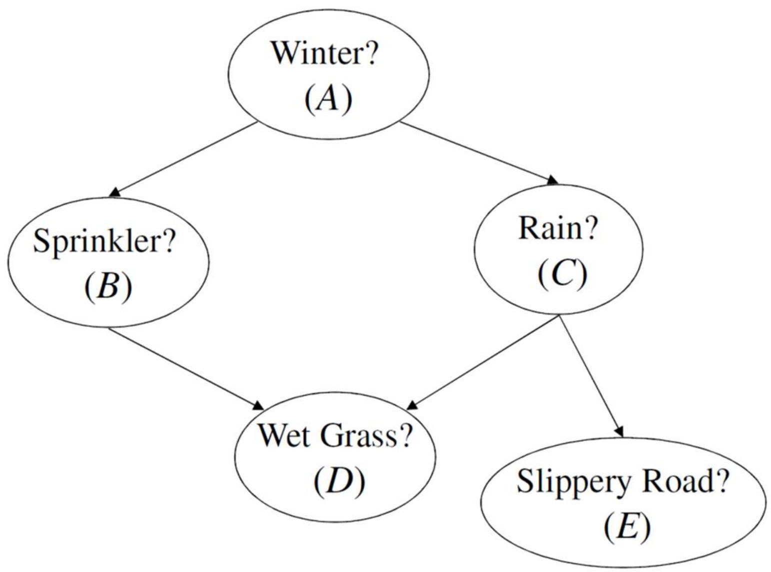
CS 6375: Machine Learning
Bayesian Networks: Inference

Vibhav Gogate

Some Slides borrowed from Adnan Darwiche and Chris Bishop

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.

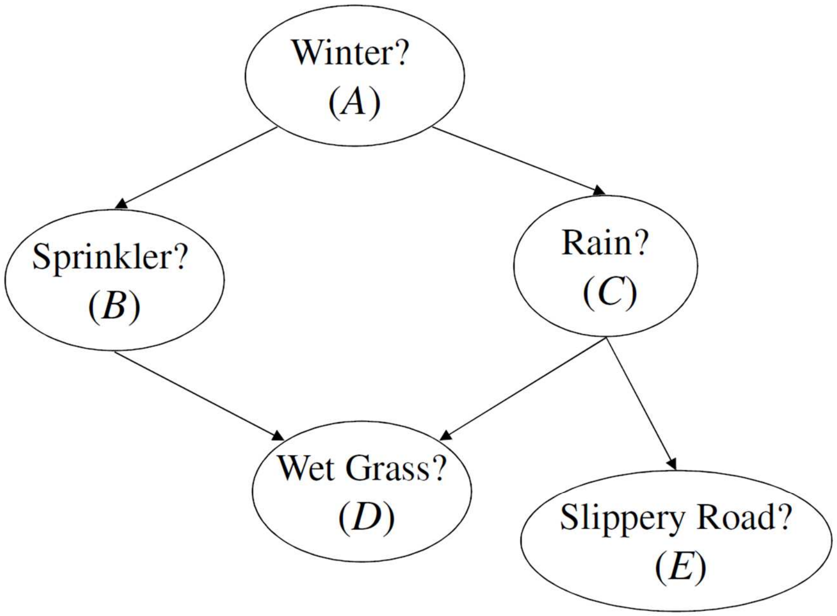
Inference Algorithms: Algorithms that take a Bayesian network as input and output an answer to the query.

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

×

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 $\mathbf{Q}$

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

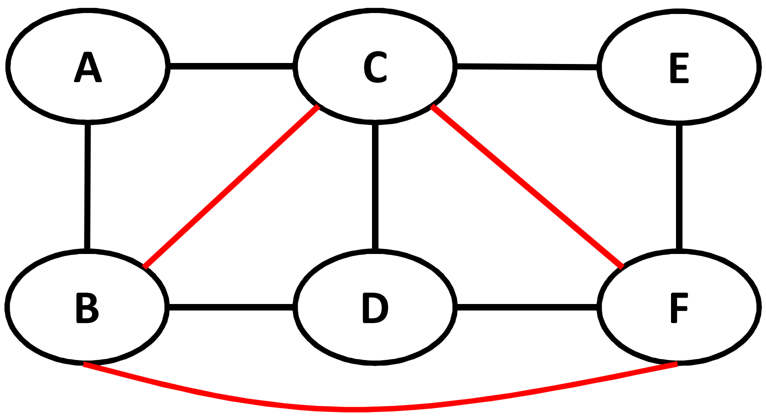
A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

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

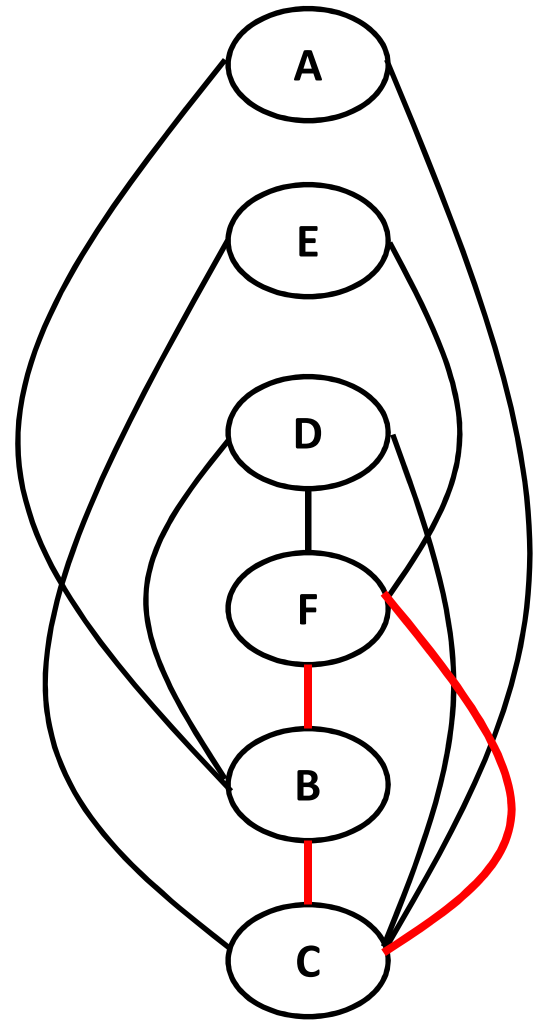
C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

Variable Elimination: Complexity

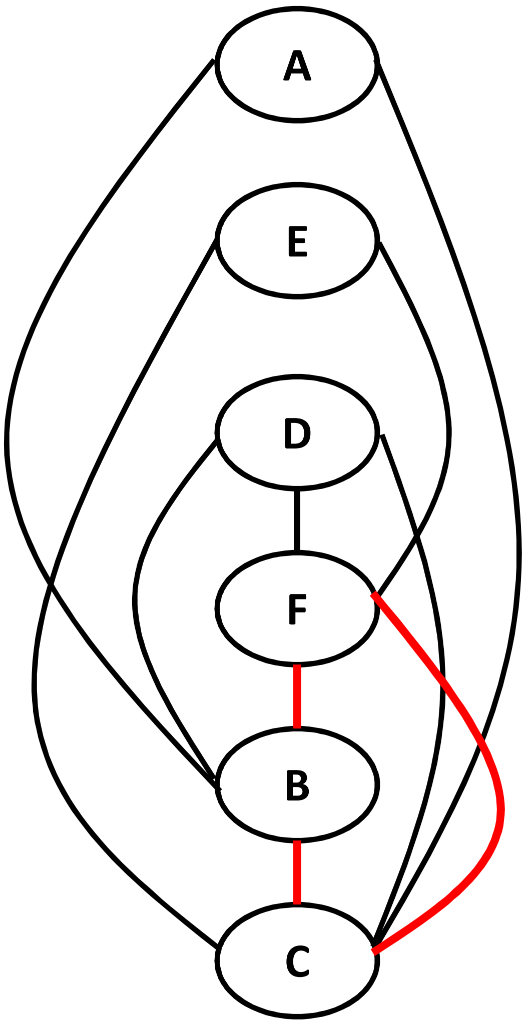
- Schematic operation on a graph



  - Process nodes in order
  - Eliminate = Connect all children of a node to each other



Variable elimination: Complexity



- Complexity of eliminating variable "i"
  - exp(children_i)
- Complexity of variable elimination:
  - nexp(max(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

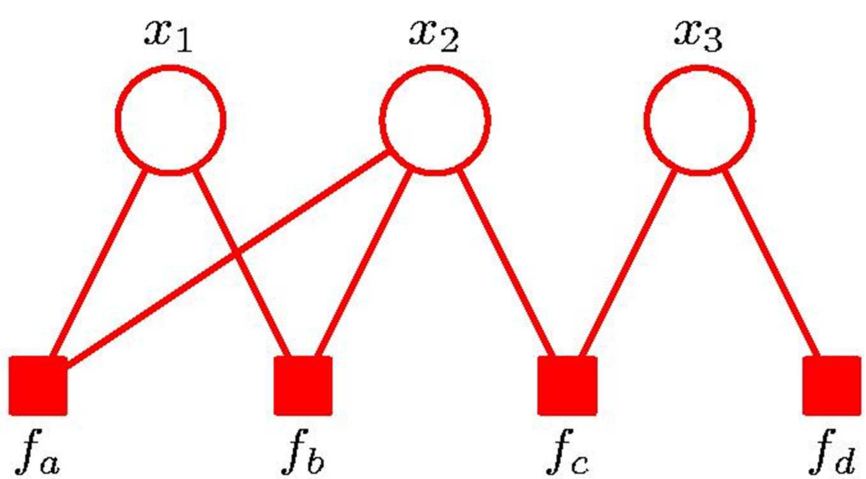
MAX_S

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

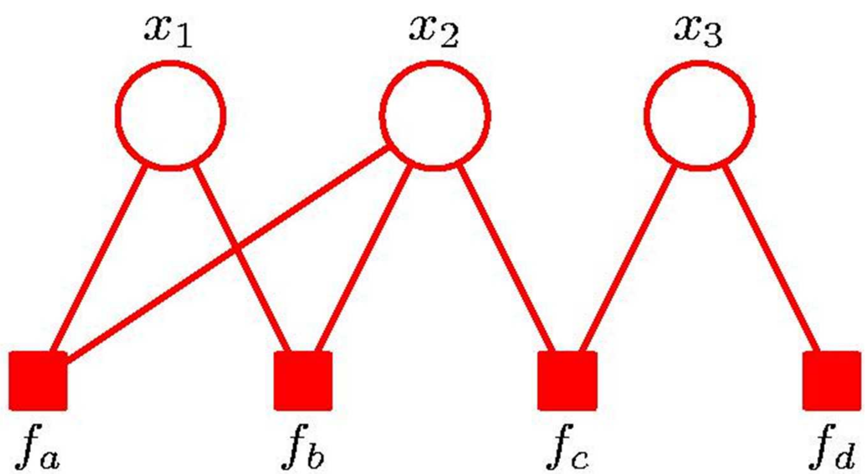
Message Passing: Factor Graphs



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

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

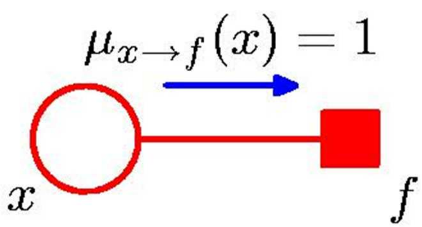
Message Passing Algorithms: Belief Propagation on Factor graphs

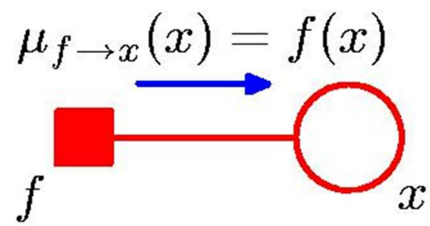


- Initialize each message
- Repeat until convergence
  - Send messages from variable nodes to factor nodes
  - Send messages from factor nodes to variable nodes
- How to construct the message from sender to receiver node?
  - At sender, multiply all incoming messages and the factor (for factor nodes only) except the message received from the receiver yielding a new factor f_S
  - Sum-out all the variables that are not in the receiver node from f_S

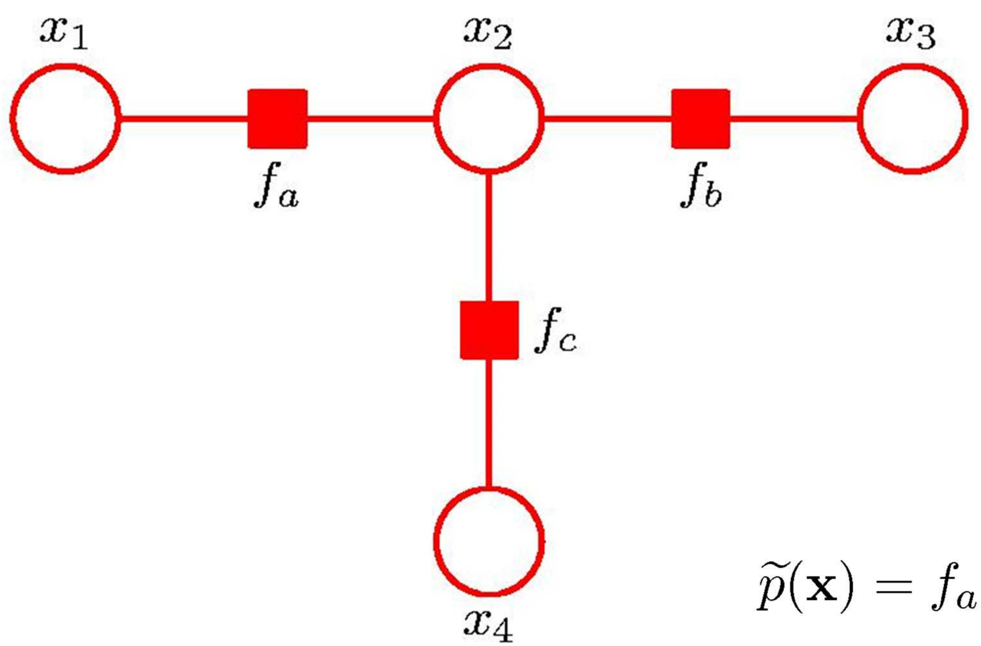
The Sum-Product Algorithm (7)

- Initialization



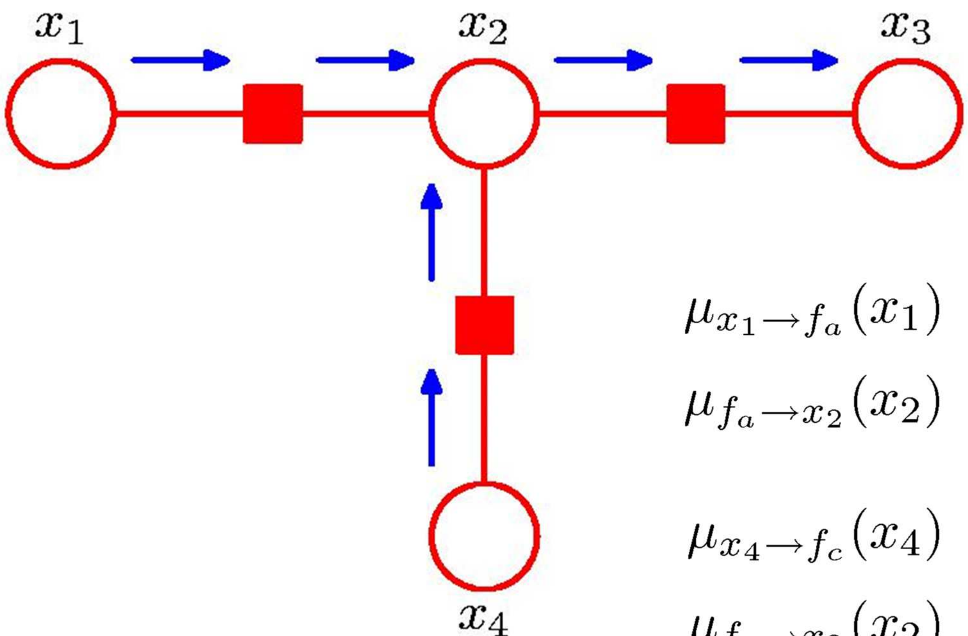


Sum-Product: Example (1)



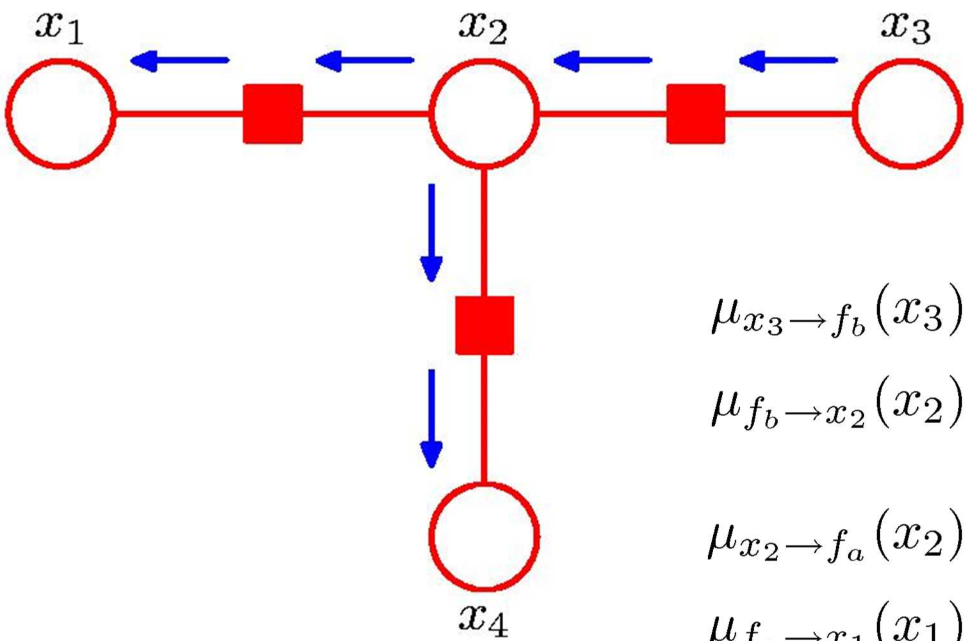
$$\widetilde{p}(\mathbf{x}) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4)$$

Sum-Product: Example (2)



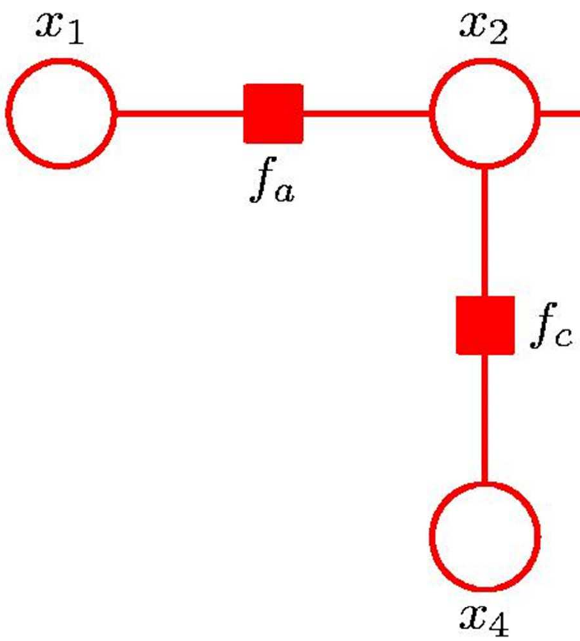
$$
\begin{aligned}
\mu_{x_1 \to f_a}(x_1) &= 1 \\
\mu_{f_a \to x_2}(x_2) &= \sum_{x_1} f_a(x_1, x_2) \\
\mu_{x_4 \to f_c}(x_4) &= 1 \\
\mu_{f_c \to x_2}(x_2) &= \sum_{x_4} f_c(x_2, x_4) \\
\mu_{x_2 \to f_b}(x_2) &= \mu_{f_a \to x_2}(x_2) \mu_{f_c \to x_2}(x_2) \\
\mu_{f_b \to x_3}(x_3) &= \sum_{x_2} f_b(x_2, x_3) \mu_{x_2 \to f_b}(x_2)
\end{aligned}
$$

Sum-Product: Example (3)



$$
\begin{aligned}
\mu_{x_3 \to f_b}(x_3) &= 1 \\
\mu_{f_b \to x_2}(x_2) &= \sum_{x_3} f_b(x_2, x_3) \\
\mu_{x_2 \to f_a}(x_2) &= \mu_{f_b \to x_2}(x_2)\mu_{f_c \to x_2}(x_2) \\
\mu_{f_a \to x_1}(x_1) &= \sum_{x_2} f_a(x_1, x_2)\mu_{x_2 \to f_a}(x_2) \\
\mu_{x_2 \to f_c}(x_2) &= \mu_{f_a \to x_2}(x_2)\mu_{f_b \to x_2}(x_2) \\
\mu_{f_c \to x_4}(x_4) &= \sum_{x_2} f_c(x_2, x_4)\mu_{x_2 \to f_c}(x_2)
\end{aligned}
$$

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 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.