CMT311 Principles of Machine Learning

Graphical Models

Angelika Kimmig
KimmigA@cardiff.ac.uk

22.11.2019

- Last week:
    - Bayesian Networks: general, graphical representation of conditional independence assumptions
- Today:
    - Graphical Models: a wider class of such representations
    - Reasoning with Graphical Models

Graphical Models

- A **Graphical model** (GM) is a graph based representation of a factorised probability distribution

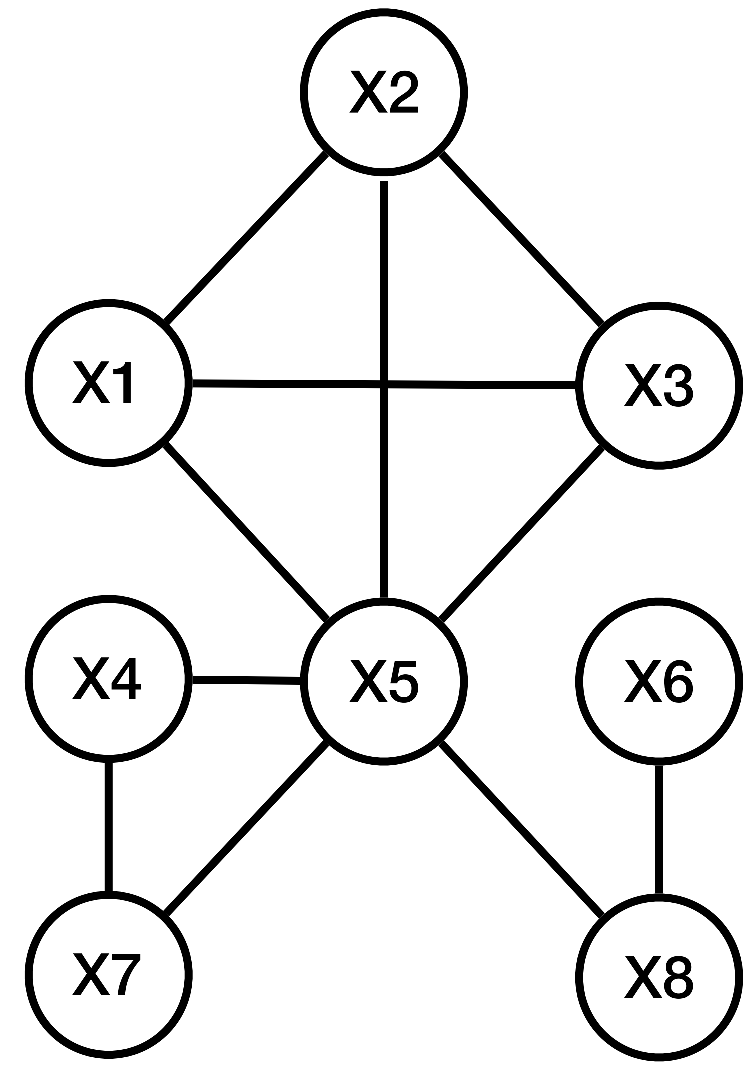
- Many different types of GMs exist, broadly falling into two categories

  - GMs mainly used for **modelling** (e.g., Bayesian networks, Markov networks, …)

  - GMs mainly used for **developing inference algorithms** (e.g., factor graphs, junction trees, …)

Background: undirected graphs

- A **graph** consists of **nodes** (vertices) and **directed** or **undirected edges**
- **undirected graph:** all edges are undirected
- **clique**: fully connected subset of nodes
- **maximal clique**: clique that is not a subset of another clique



Markov Networks

- A **potential** is a function from a set of random variables to the non-negative numbers.

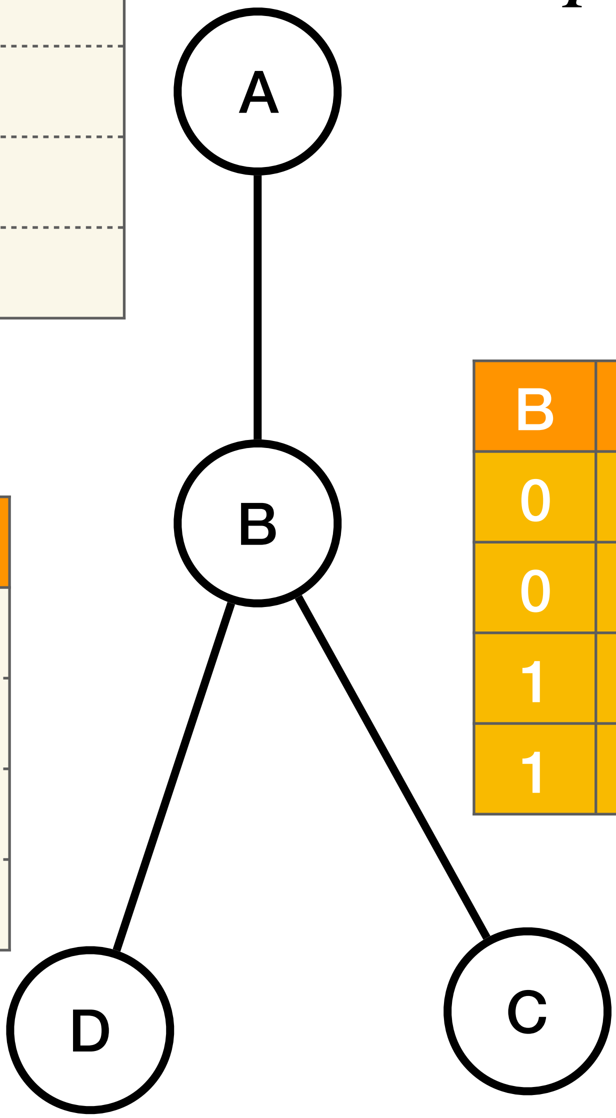
- A **Markov network** (MN) is an undirected graph in which each node corresponds to a random variable, and a potential ψ is defined on each (maximal) clique.

- The joint probability distribution represented by an MN is the normalised product of all clique potentials in the MN:
  $P(X_1, \ldots, X_n) = \frac{1}{Z}\prod_{i=1}^{m}\psi_i(\mathcal{X}_i)$, where $\mathcal{X}_i$ denotes the set of variables in the i-th clique, and the normalisation constant is $Z = \sum_{X_1,\ldots,X_n}\prod_{i=1}^{m}\psi_i(\mathcal{X}_i)$

MN: example

A	B	$f_1(A,B)$
0	0	10
0	1	1
1	0	1
1	1	10

$$P(A,B,C,D) \propto f_1(A,B) \cdot f_2(B,C) \cdot f_3(B,D)$$

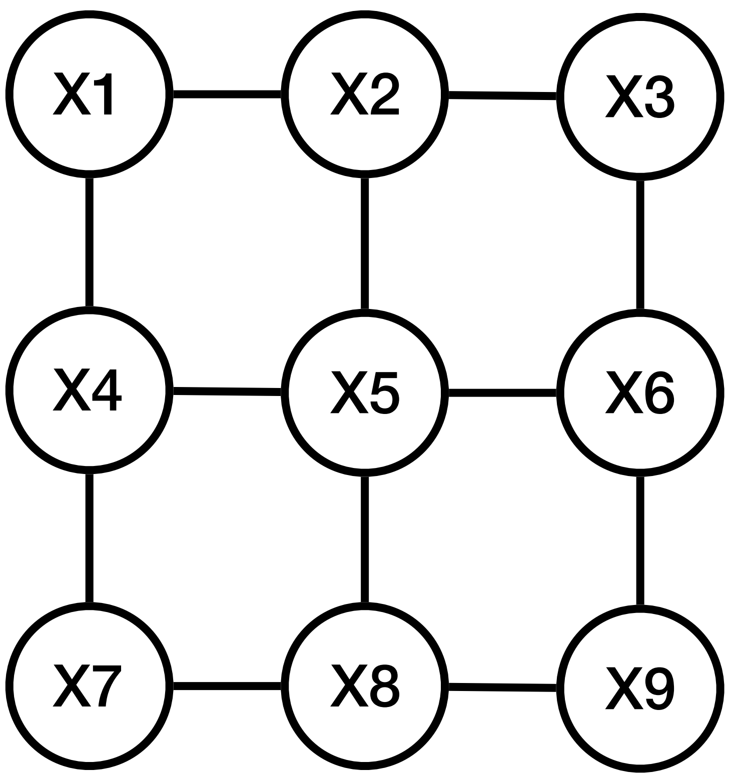


B	C	$f_2(B,C)$
0	0	1
0	1	10
1	0	10
1	1	1

B	D	$f_3(B,D)$
0	0	10
0	1	1
1	0	1
1	1	10

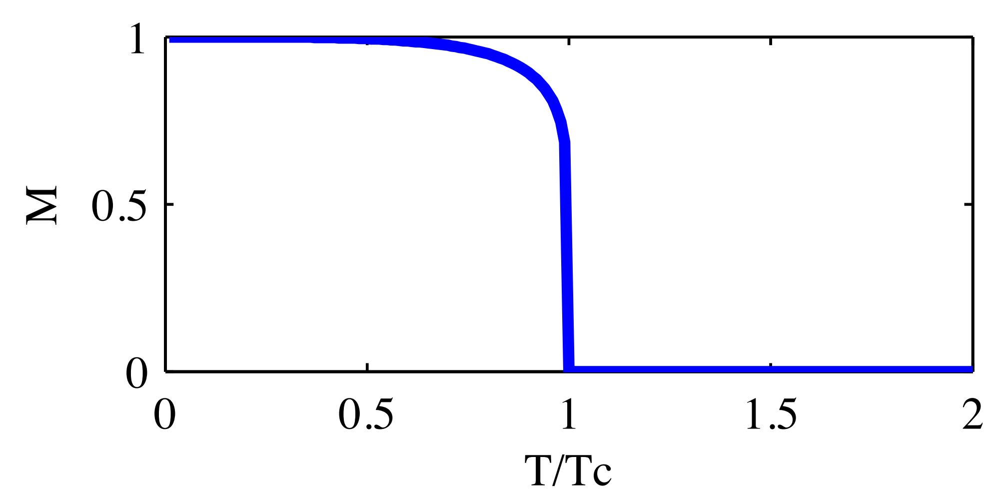
$$Z = \sum_{(a,b,c,d)\in\{0,1\}^4} f_1(A=a,B=b) \cdot f_2(B=b,C=c) \cdot f_3(B=b,D=d)$$

The Ising model



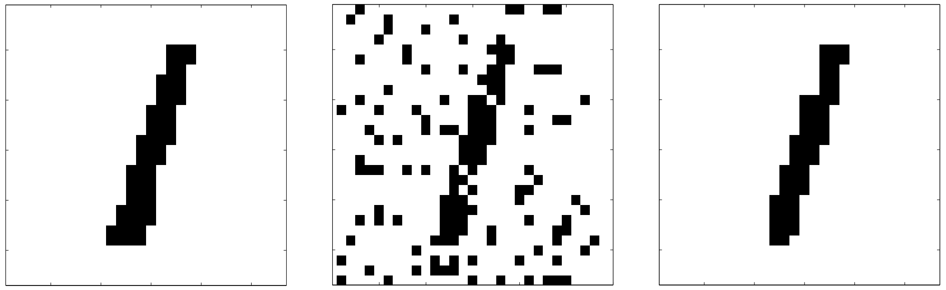
- well-known in physics of magnetic systems
- all variables take values +1 or -1
- for each pair of nodes Xi and Xj connected by an edge $\phi_{ij}(X_i, X_j) = e^{-\frac{1}{2T}(X_i - X_j)^2}$
- the temperature T is a parameter controlling how much neighbouring nodes are encouraged to take the same value

$$M = |\sum_{i=1}^{N} x_i|/N$$

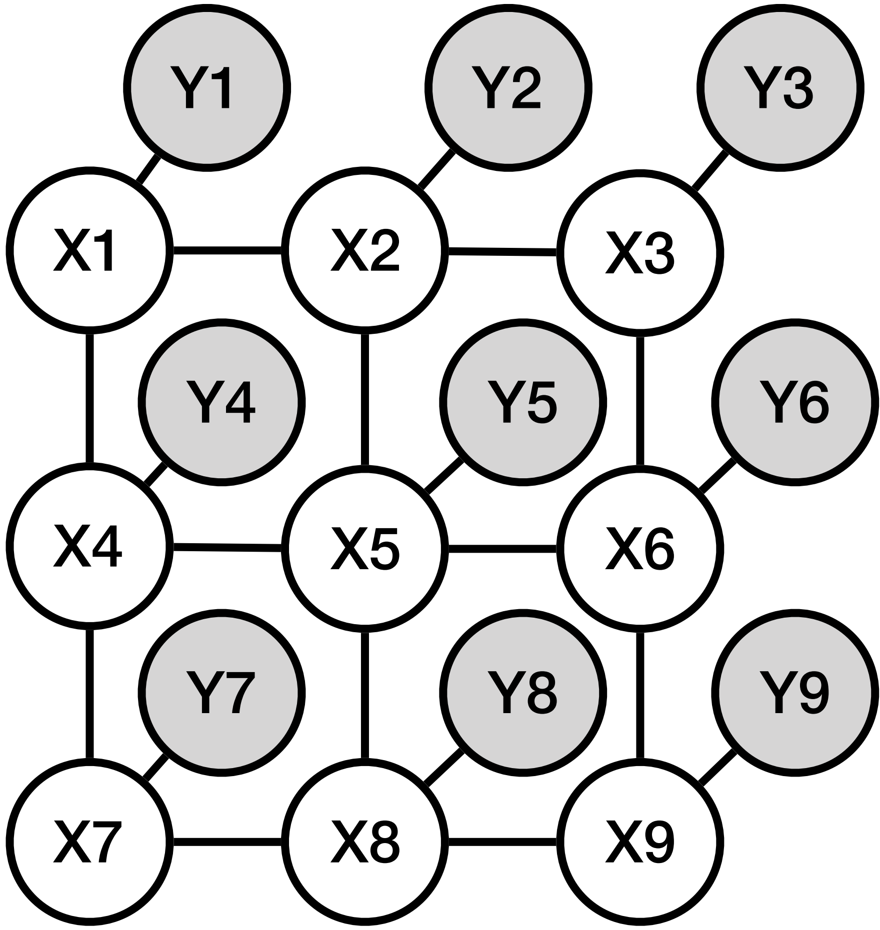


MN for image cleaning

- Task: recover a binary picture from a corrupted version



- clean pixels $X_i \in \{-1, +1\}$ (unobserved)
- corrupted pixels $Y_i \in \{-1, +1\}$ (observed)
- $\phi(Y_i, X_i) = e^{\gamma X_i Y_i}$ encourage X_i and Y_i to be similar
- $\psi(X_i, X_j) = e^{\beta X_i X_j}$ for neighbouring X_i and X_j encourage the image to be smooth
- $$P(X_1, \ldots, X_n, Y_1, \ldots, Y_n) \propto \prod_i \phi(Y_i, X_i) \prod_{i \sim j} \psi(X_i, X_j)$$



Global Markov property

- Let $\mathscr{X}$, $\mathscr{Y}$ and $\mathscr{Z}$ be disjoint sets of random variables
- $\mathscr{Z}$ **separates** $\mathscr{X}$ and $\mathscr{Y}$ if every path from any variable in $\mathscr{X}$ to any variable in $\mathscr{Y}$ passes through $\mathscr{Z}$
- **Global Markov Property**: if $\mathscr{Z}$ separates $\mathscr{X}$ and $\mathscr{Y}$, then $\mathscr{X} \perp\!\!\!\perp \mathscr{Y} \mid \mathscr{Z}$

Markov Properties

- write $X_{\backslash i}$ for the set of all variables except X_i, and $ne(X_i)$ for the set of all variables connected to X_i by an edge
- **Local Markov property:** $P(X_i \mid X_{\backslash i}) = P(X_i \mid ne(X_i))$
- **Pairwise Markov property** (follows from local one): if there is no edge between X and Y, then $X \perp\!\!\!\perp Y \mid \mathcal{X} \backslash \{X, Y\}$
- If all potentials are **positive**,
  - the local, pairwise and global properties are all equivalent
  - these properties hold if and only if the distribution has the factorised form

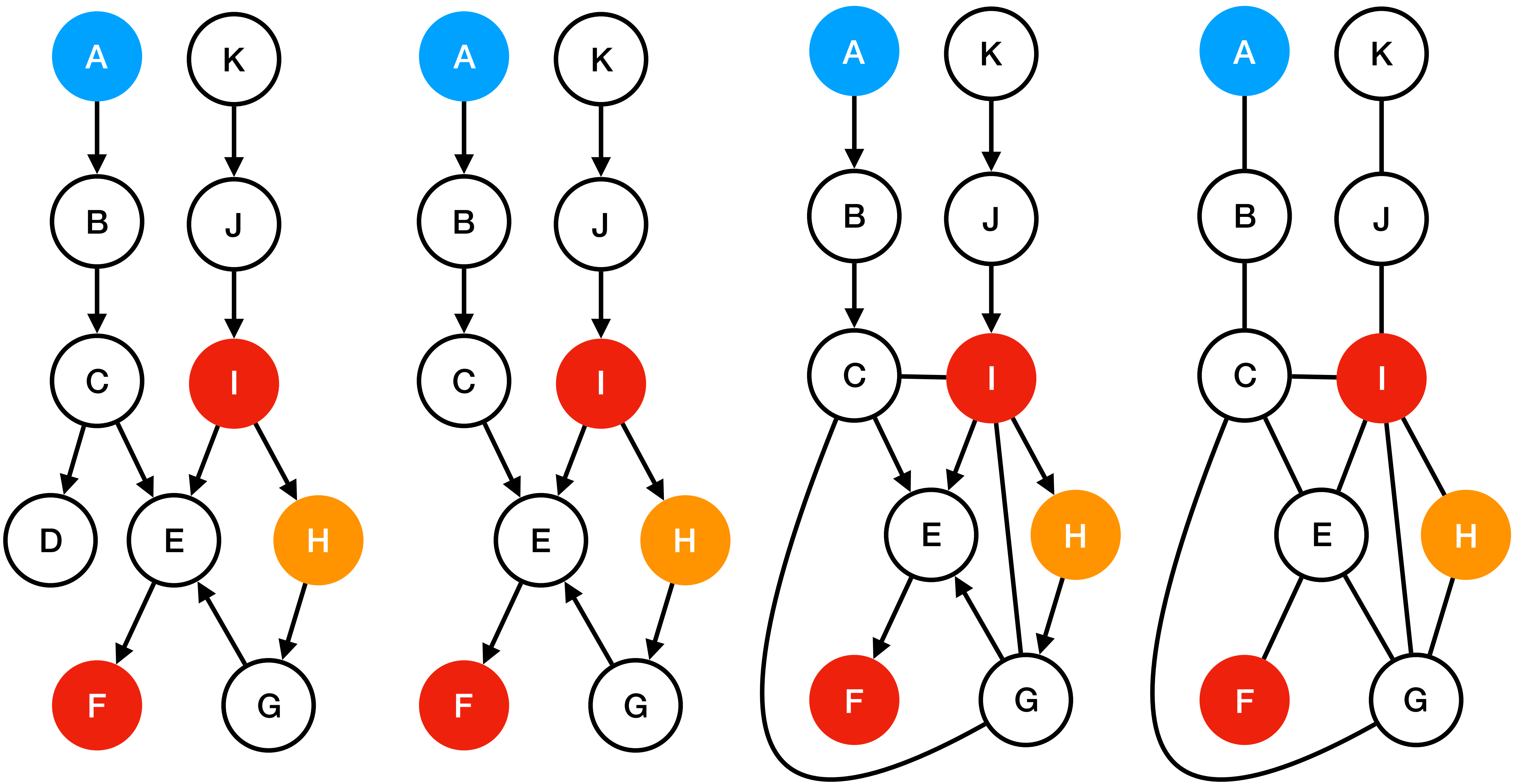
Alternative Independence Check for BNs

- to check whether $\mathscr{X} \perp\!\!\!\perp \mathscr{Y} \mid \mathscr{Z}$ in a BN
  - remove any nodes that are neither in $\mathscr{X} \cup \mathscr{Y} \cup \mathscr{Z}$ nor an ancestor of a node in $\mathscr{X} \cup \mathscr{Y} \cup \mathscr{Z}$ as well as all edges involving removed nodes
  - add an edge between any two parents of the same child
  - turn the graph into its skeleton
  - if $\mathscr{Z}$ separates $\mathscr{X}$ and $\mathscr{Y}$ in this graph, then $\mathscr{X} \perp\!\!\!\perp \mathscr{Y} \mid \mathscr{Z}$

Example

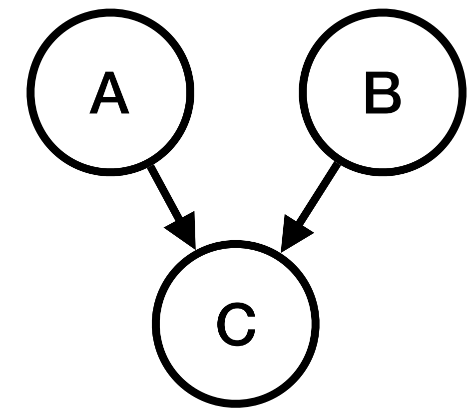
$A \perp\!\!\!\perp H \mid \{F, I\}$?

no!



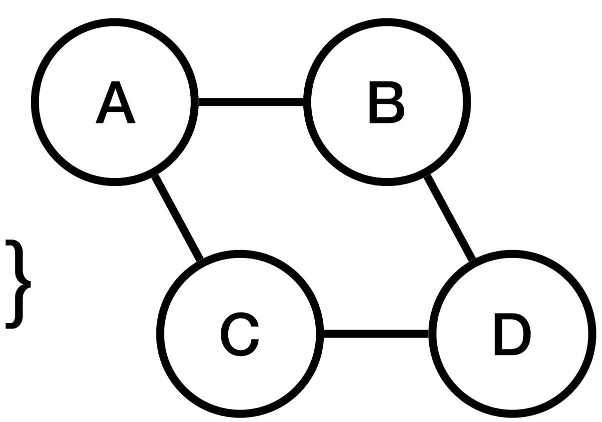
Expressiveness: BNs vs MNs

$A \perp\!\!\!\perp B$



- any BN can be turned into an MN:
  - just use the CPTs as potentials
  - however, independence information may be lost in the graph structure
- not all MNs can be turned into BNs with the same link structure

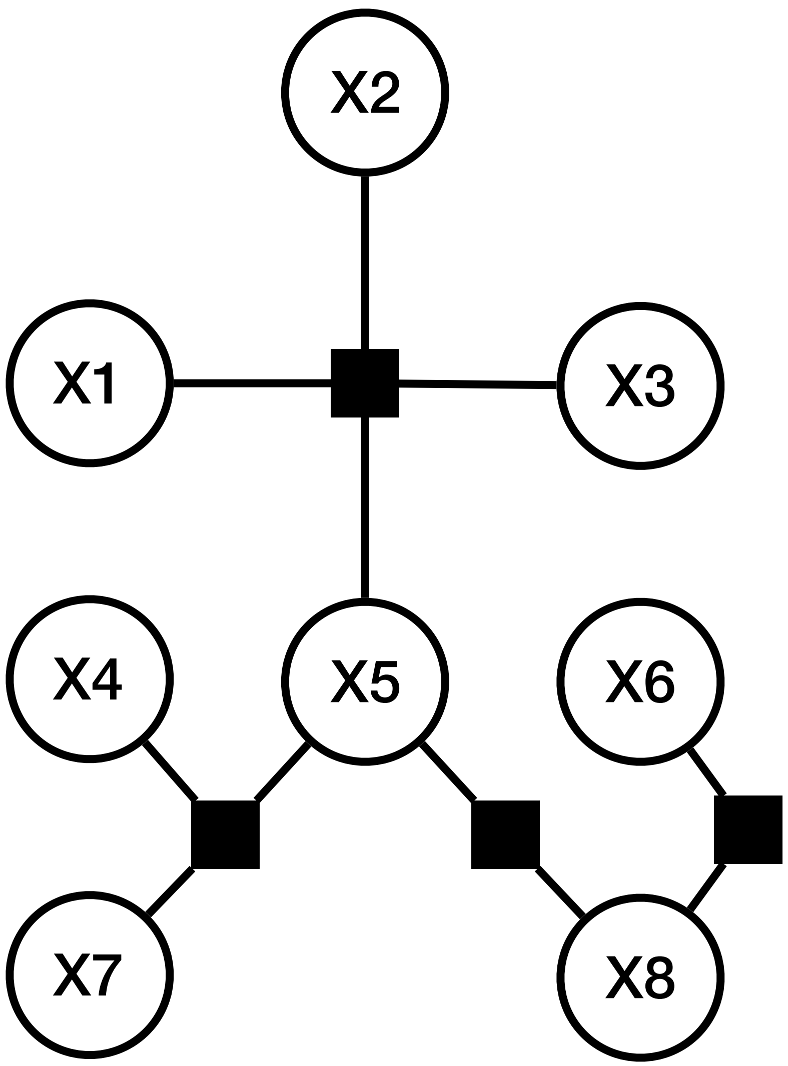
$A \perp\!\!\!\perp D | \{B,C\}$



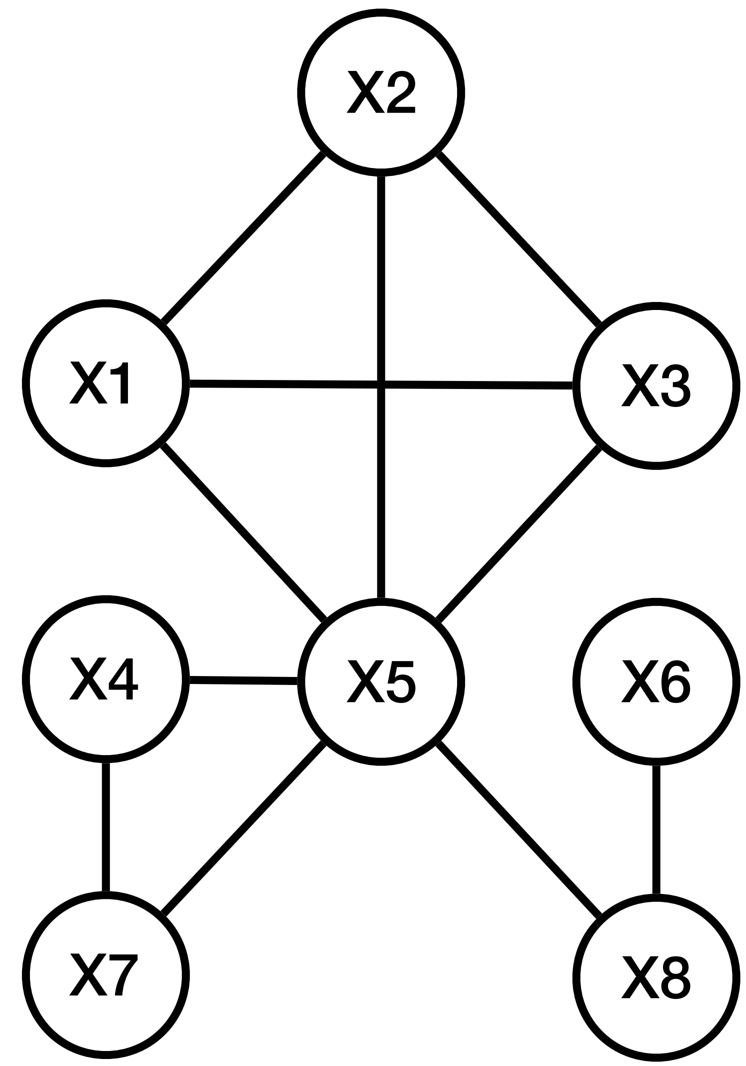
Factor Graphs

- The factor graph (FG) of a function $f(X_1, \ldots, X_n) = \prod_{i=1}^{m} \psi_i(\mathscr{X}_i)$ consists of
  - a factor node (represented as square) for every ψ_i
  - a variable node (represented as circle) for every X_j
  - an undirected edge between X_j and ψ_i for every $X_j \in \mathscr{X}_i$

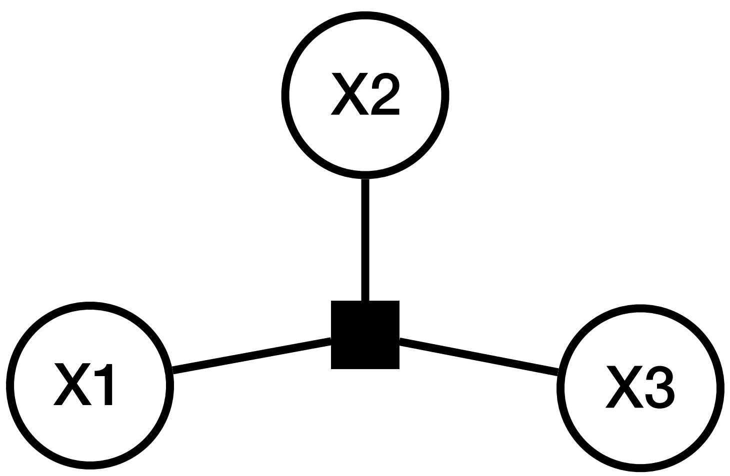
Example



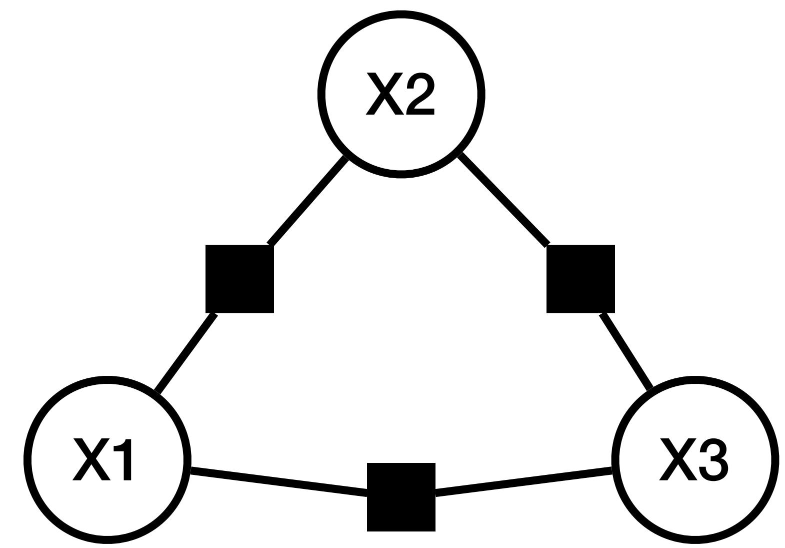
corresponds to the MN



Example

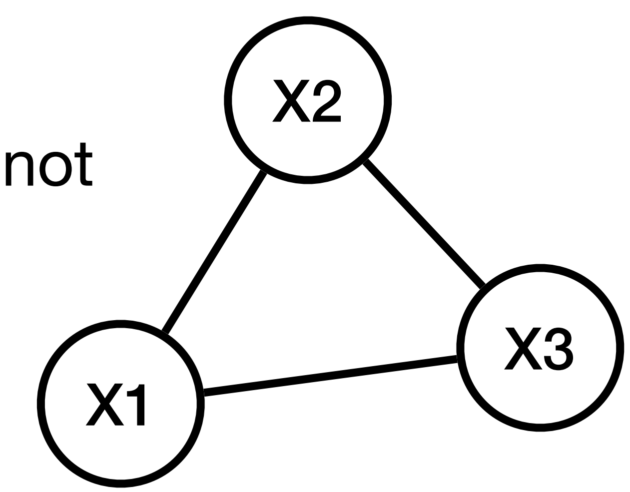


$\psi(X_1, X_2, X_3)$



$\psi_1(X_1, X_2)\psi_2(X_2, X_3)\psi_3(X_3, X_1)$

this distinction of the form cannot be made by an MN:



Graphical Models

- A **Graphical model** (GM) is a graph based representation of a factorised probability distribution

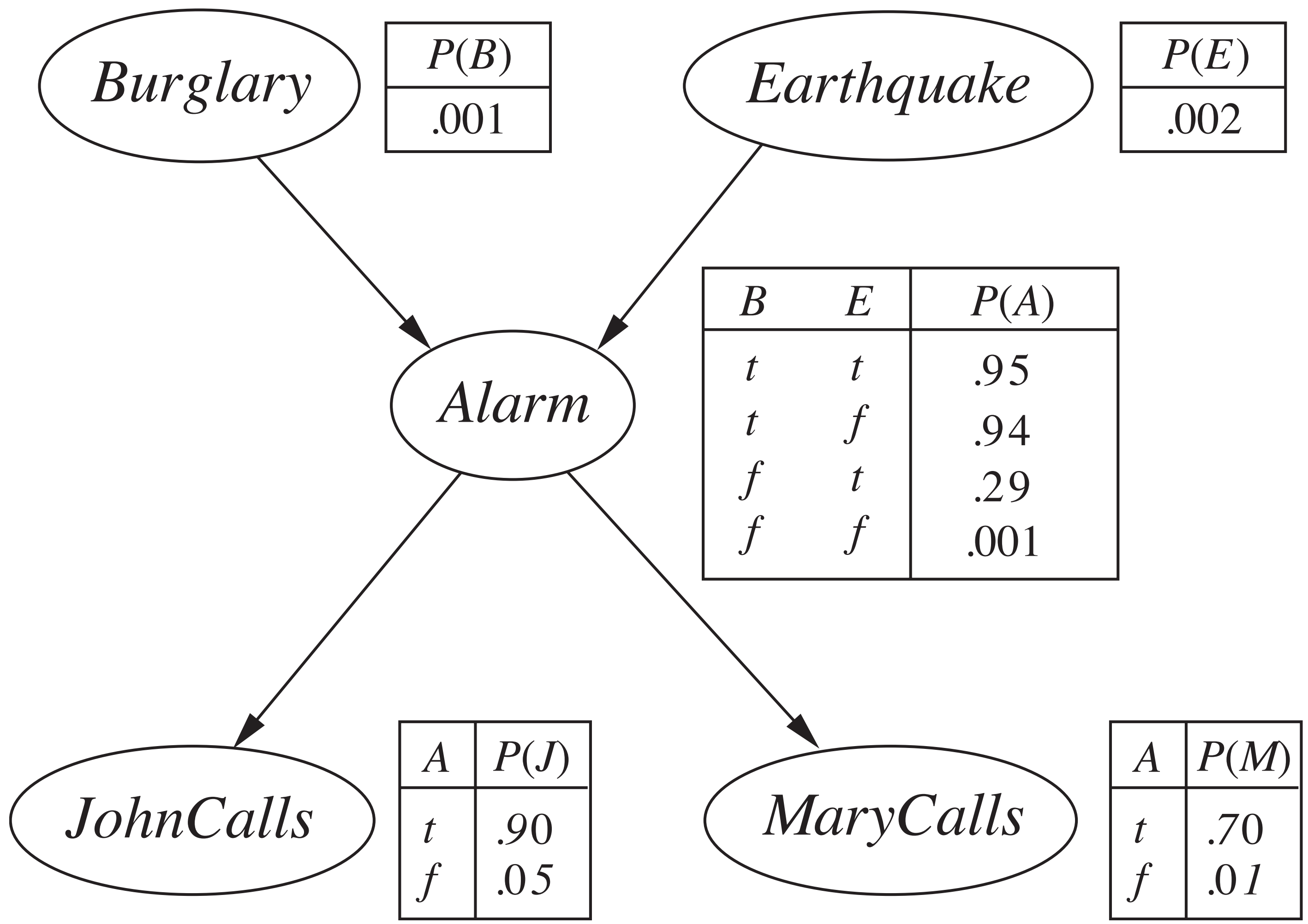
- Many different types of GMs exist, broadly falling into two categories

  - GMs mainly used for **modelling** (e.g., Bayesian networks, Markov networks, …)

  - GMs mainly used for **developing inference algorithms** (e.g., factor graphs, junction trees, …)

Judea Pearl's Alarm network

You have a new burglary alarm that is fairly reliable at detecting a burglary, but also responds to earthquakes. Your neighbours, Mary and John, promise to call you if they hear the alarm sounding.



$P(B)$
.001

$P(E)$
.002

B	E	$P(A)$
t	t	.95
t	f	.94
f	t	.29
f	f	.001

A	$P(J)$
t	.90
f	.05

A	$P(M)$
t	.70
f	.01

[figure: Russell& Norvig]

BN Inference

- A Bayesian network defines a joint probability distribution

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$

We'll focus on computing conditional probabilities

- **Inference** is the task of using the distribution to answer questions, such as
  - **computing** the **conditional probability** distribution of one variable given certain observations

    e.g., P(Earthquake | MaryCalls=t)
  - **finding** the **most probable world** where certain observations hold

    e.g., if we know MaryCalls=t, what is the most likely combination of values for the other four variables?
  - **finding** the **most likely value** of one variable given certain observations

    e.g., if we know MaryCalls=t, is it more likely that JohnCalls=t or JohnCalls=f?

Example

- What is the probability of a burglary given both John and Mary call?
- i.e., what is the conditional distribution P(B|J=t,M=t)?
- by definition,

$$P(B \mid J = t, M = t) = \frac{P(B, J = t, M = t)}{P(J = t, M = t)} = \frac{P(B, J = t, M = t)}{P(B = t, J = t, M = t) + P(B = f, J = t, M = t)}$$

- we'll focus on computing P(B=t,J=t,M=t):

$$P(B = t, J = t, M = t) = \sum_{E} \sum_{A} P(E)P(B = t)P(A \mid E, B = t)P(J = t \mid A)P(M = t \mid A)$$

Evaluate the sum by looping over all combinations of values for E and A

$$P(B=t, J=t, M=t) = \sum_E \sum_A P(E)P(B=t)P(A \mid E, B=t)P(J=t \mid A)P(M=t \mid A)$$

$$= P(E=t)P(B=t)P(A=t \mid E=t, B=t)P(J=t \mid A=t)P(M=t \mid A=t)$$

$$+P(E=t)P(B=t)P(A=f \mid E=t, B=t)P(J=t \mid A=f)P(M=t \mid A=f)$$

$$+P(E=f)P(B=t)P(A=t \mid E=f, B=t)P(J=t \mid A=t)P(M=t \mid A=t)$$

$$+P(E=f)P(B=t)P(A=f \mid E=f, B=t)P(J=t \mid A=f)P(M=t \mid A=f)$$

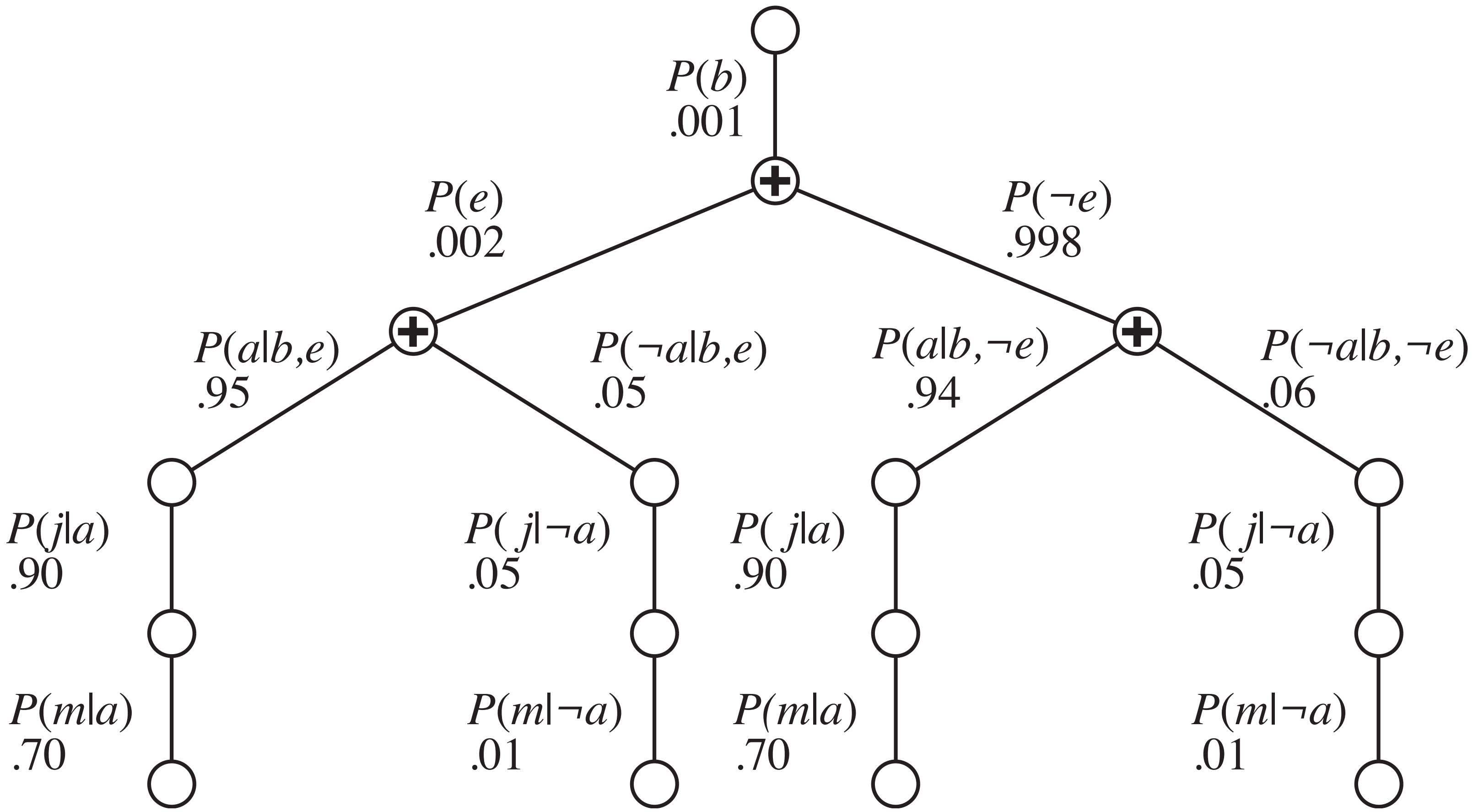
Correct, but we can avoid some work: exploit distributivity

$$P(B=t, J=t, M=t) = P(B=t) \sum_E \sum_A P(E)P(A \mid E, B=t)P(J=t \mid A)P(M=t \mid A)$$

$$= P(B=t) \sum_E P(E) \sum_A P(A \mid E, B=t)P(J=t \mid A)P(M=t \mid A)$$

$$P(B=t, J=t, M=t) = P(B=t) \sum_{E} P(E) \sum_{A} P(A \mid E, B=t) P(J=t \mid A) P(M=t \mid A)$$

Left-to-right evaluation corresponds to depth-first traversal of tree:



[notation: for variable X, write x for $X = t$ and $\neg x$ for $X = f$]

Correct, but we can avoid some work: start at bottom and cache intermediate results

[figure: Russell& Norvig]

Variable Elimination

$$P(B, J=t, M=t) = \underbrace{\underbrace{P(B)}_{f_B(B)} \underbrace{\sum_E \underbrace{P(E)}_{f_E(E)} \underbrace{\sum_A \underbrace{P(A \mid E, B)}_{f_A(A,B,E)} \underbrace{P(J=t \mid A)}_{f_J(A)} \underbrace{P(M=t \mid A)}_{f_M(A)}}_{f_{\overline{A}JM}(B,E)}}_{f_{\overline{E}\overline{A}JM}(B)}}_{P(B, J=t, M=t)}$$

$$f_{\overline{A}JM}(B,E) = \sum_a f_A(A=a, B, E) \times f_J(A=a) \times f_M(A=a)$$

$$= f_A(A=t, B, E) \times f_J(A=t) \times f_M(A=t) + f_A(A=f, B, E) \times f_J(A=f) \times f_M(A=f)$$

$$f_{\overline{E}\overline{A}JM}(B) = f_E(E=t) \times f_{\overline{A}JM}(B, E=t) + f_E(E=f) \times f_{\overline{A}JM}(B, E=f)$$

$$P(B, J=t, M=t) = f_B(B) \times f_{\overline{E}\overline{A}JM}(B)$$

Multiplying Factors

$$f_1(X_1, \ldots, X_j, Y_1, \ldots, Y_k) \times f_2(Y_1, \ldots, Y_k, Z_1, \ldots, Z_l)$$
$$= f(X_1, \ldots, X_j, Y_1, \ldots, Y_k, Z_1, \ldots, Z_l)$$

Example

$f_1(A, B) \times f_2(B, C)$

A	B	f_1(A,B)
0	0	0.1
0	1	0.2
1	0	0.3
1	1	0.4

x

B	C	f_2(B,C)
0	0	0.5
0	1	0.6
1	0	0.7
1	1	0.8

=

A	B	C	f(A,B,C)
0	0	0	0.1*0.5
0	0	1	0.1*0.6
0	1	0	0.2*0.7
0	1	1	0.2*0.8
1	0	0	0.3*0.5
1	0	1	0.3*0.6
1	1	0	0.4*0.7
1	1	1	0.4*0.8

Variable Elimination

$$P(B, J=t, M=t) = \underbrace{P(B)}_{f_B(B)} \sum_E \underbrace{P(E)}_{f_E(E)} \sum_A \underbrace{P(A \mid E, B)}_{f_A(A,B,E)} \underbrace{P(J=t \mid A)}_{f_J(A)} \underbrace{P(M=t \mid A)}_{f_M(A)}$$

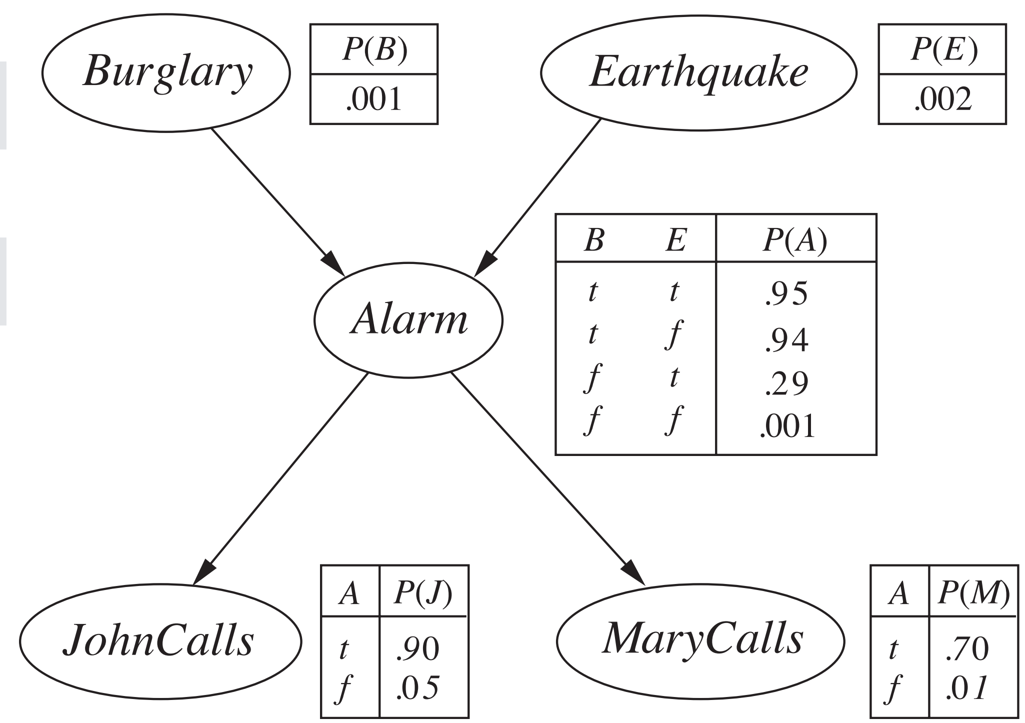
B	$f_B(B)$
t	0.001
f	0.999

E	$f_E(E)$
t	0.002
f	0.998

A	B	E	$f_A(A,B,E)$
t	t	t	0.95
t	t	f	0.94
t	f	t	0.29
t	f	f	0.001
f	t	t	0.05
f	t	f	0.06
f	f	t	0.71
f	f	f	0.999

A	$f_J(A)$
t	0.9
f	0.05

A	$f_M(A)$
t	0.7
f	0.01



[figure: Russell& Norvig]

Variable Elimination

$$P(B, J=t, M=t) = \underbrace{P(B)}_{f_B(B)} \sum_E \underbrace{P(E)}_{f_E(E)} \sum_A \underbrace{\underbrace{P(A \mid E, B)}_{f_A(A,B,E)} \underbrace{P(J=t \mid A)}_{f_J(A)} \underbrace{P(M=t \mid A)}_{f_M(A)}}$$

$$f_{\overline{A}JM}(B,E) = \sum_a \boxed{f_A(A=a,B,E) \times f_J(A=a) \times f_M(A=a)}$$

A	B	E	$f_A(A,B,E)$
t	t	t	0.95
t	t	f	0.94
t	f	t	0.29
t	f	f	0.001
f	t	t	0.05
f	t	f	0.06
f	f	t	0.71
f	f	f	0.999

x

A	$f_J(A)$
t	0.9
f	0.05

x

A	$f_M(A)$
t	0.7
f	0.01

=

A	B	E	
t	t	t	0.95*0.9*0.7
t	t	f	0.94*0.9*0.7
t	f	t	0.29*0.9*0.7
t	f	f	0.001*0.9*0.7
f	t	t	0.05*0.05*0.01
f	t	f	0.06*0.05*0.01
f	f	t	0.71*0.05*0.01
f	f	f	0.999*0.05*0.01

Variable Elimination

$$P(B, J=t, M=t) = \underbrace{P(B)}_{f_B(B)} \sum_E \underbrace{P(E)}_{f_E(E)} \sum_A \underbrace{P(A \mid E, B)}_{f_A(A,B,E)} \underbrace{P(J=t \mid A)}_{f_J(A)} \underbrace{P(M=t \mid A)}_{f_M(A)}$$

$$f_{\bar{A}JM}(B,E) = \sum_a \boxed{f_A(A=a,B,E) \times f_J(A=a) \times f_M(A=a)}$$

A	B	E	
t	t	t	0.95*0.9*0.7
t	t	f	0.94*0.9*0.7
t	f	t	0.29*0.9*0.7
t	f	f	0.001*0.9*0.7
f	t	t	0.05*0.05*0.01
f	t	f	0.06*0.05*0.01
f	f	t	0.71*0.05*0.01
f	f	f	0.999*0.05*0.01

summing out A:

B	E	$f_{\bar{A}JM}(B,E)$
t	t	0.95*0.9*0.7+0.05*0.05*0.01=0.598525
t	f	0.94*0.9*0.7+0.06*0.05*0.01=0.59223
f	t	0.29*0.9*0.7+0.71*0.05*0.01=0.183055
f	f	0.001*0.9*0.7+0.999*0.05*0.01=0.0011295

Variable Elimination

$$P(B, J=t, M=t) = \underbrace{P(B)}_{f_B(B)} \sum_E \underbrace{P(E)}_{f_E(E)} \sum_A \underbrace{P(A \mid E, B) P(J=t \mid A) P(M=t \mid A)}_{f_{\bar{A}JM}(B,E)}$$

$$\underbrace{\sum_E P(E) \sum_A P(A \mid E, B) P(J=t \mid A) P(M=t \mid A)}_{f_{\bar{E}\bar{A}JM}(B)}$$

E	$f_E(E)$
t	0.002
f	0.998

x

B	E	$f_{\bar{A}JM}(B,E)$
t	t	0.598525
t	f	0.59223
f	t	0.183055
f	f	0.0011295

=

B	E	
t	t	0.598525*0.002
t	f	0.59223*0.998
f	t	0.183055*0.002
f	f	0.0011295*0.998

summing out E:

B	$f_{\bar{E}\bar{A}JM}(B)$
t	0.598525*0.002+0.59223*0.998=0.59224259
f	0.183055*0.002+0.0011295*0.998=0.001493351

Variable Elimination

$$P(B, J=t, M=t) = \underbrace{\underbrace{P(B)}_{f_B(B)} \underbrace{\sum_E P(E) \sum_A P(A \mid E, B) P(J=t \mid A) P(M=t \mid A)}_{f_{\overline{EA}JM}(B)}}_{P(B, J=t, M=t)}$$

B	$f_B(B)$
t	0.001
f	0.999

x

B	$f_{\overline{EA}JM}(B)$
t	0.59224259
f	0.001493351

=

B	P(B,J=t,M=t)
t	0.59224259*0.001=0.00059224259
f	0.001493351*0.999=0.001491857649

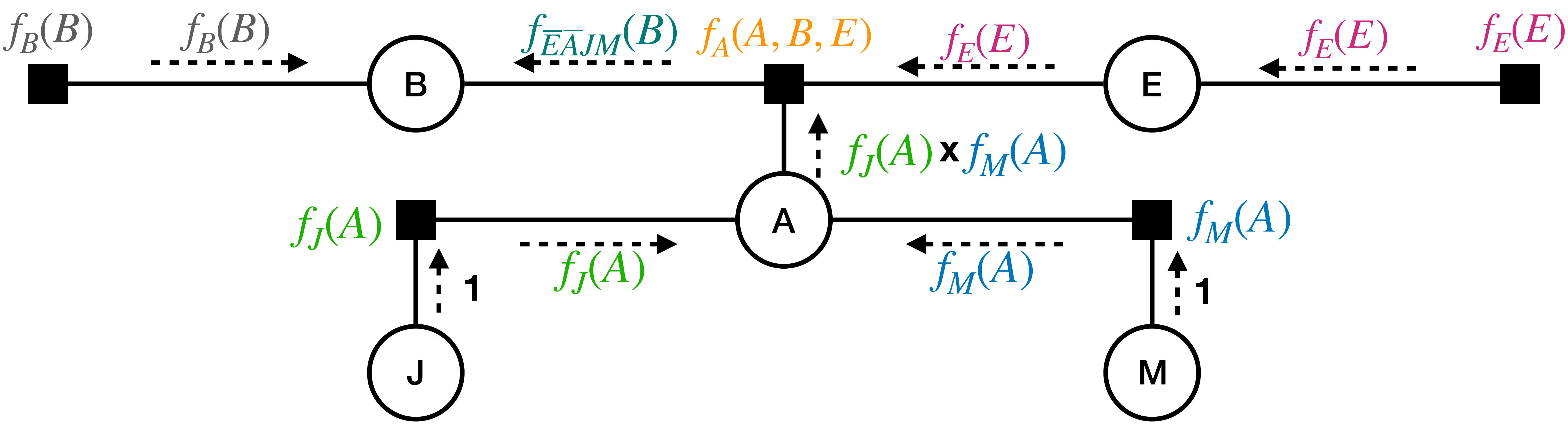
normalisation gives conditional probabilities:

$$P(B=t \mid J=t, M=t) = \frac{0.00059224259}{0.00059224259 + 0.001491857649} = 0.2842$$

$$P(B=f \mid J=t, M=t) = \frac{0.001491857649}{0.00059224259 + 0.001491857649} = 0.7158$$

Variable Elimination as Message Passing

$$P(B, J=t, M=t) = \underbrace{\underbrace{P(B)}_{f_B(B)} \underbrace{\sum_E \underbrace{P(E)}_{f_E(E)} \underbrace{\sum_A \underbrace{P(A \mid E, B)}_{f_A(A,B,E)} \underbrace{P(J=t \mid A)}_{f_J(A)} \underbrace{P(M=t \mid A)}_{f_M(A)}}_{f_{\overline{A}JM}(B,E)}}_{f_{\overline{E}\overline{A}JM}(B)}}_{P(B, J=t, M=t)}$$



Sum-Product Algorithm

- the sum-product algorithm uses message passing to compute marginals of all variables on factor graphs without loops
- as any BN or MN can be turned into a factor graph, same algorithm works for both types of models
- sometimes also called belief propagation
- two types of messages:
  - from variables to factors
  - from factors to variables
- node X can send message to neighbour Y only after having received messages from all other neighbours

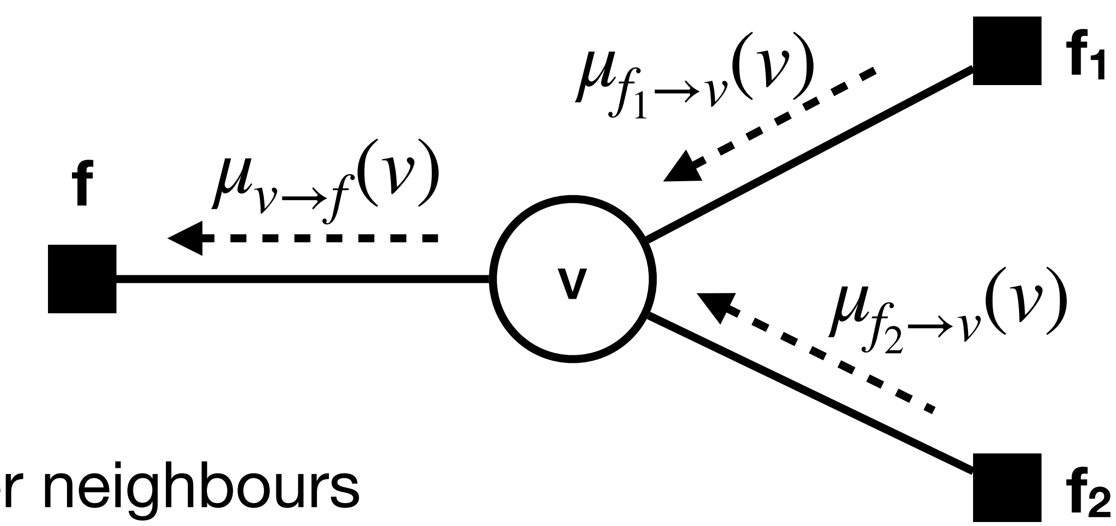
Sum-Product Algorithm

- pick one node as the root node
- initialisation:
  - set messages from leaf factors to their factor
  - set messages from leaf variables to one
- step 1: propagate messages from leaves to root
- step 2: propagate messages from root to leaves

Sum-Product Algorithm

variable to factor messages

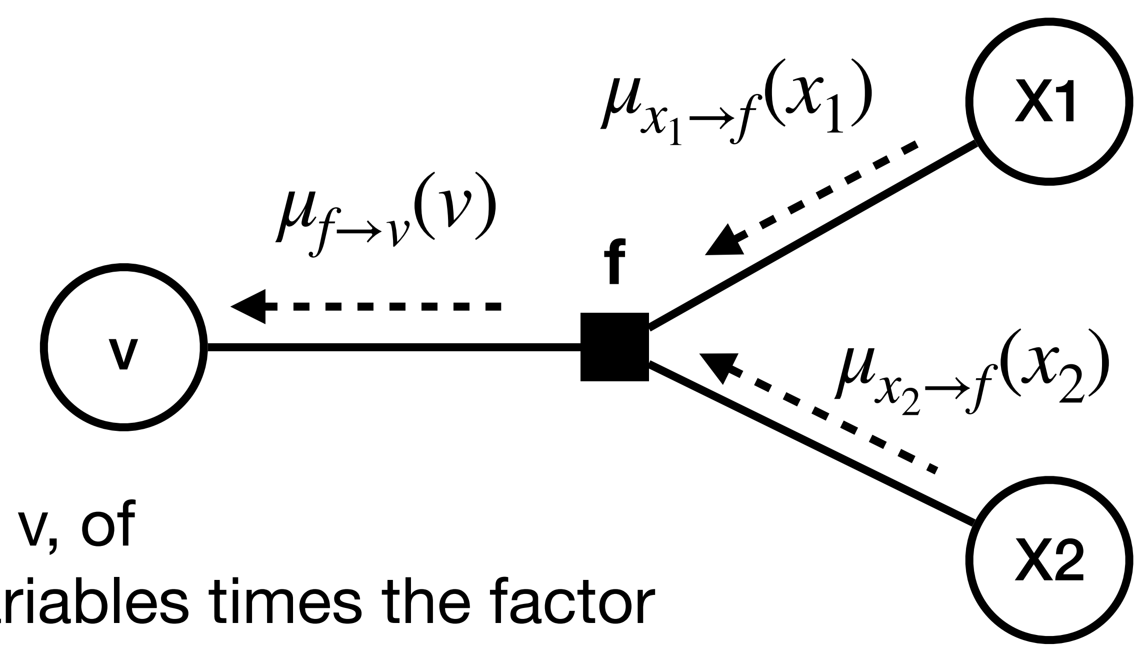
$$\mu_{v\to f}(v) = \prod_{f_i \in ne(v)\setminus\{f\}} \mu_{f_i\to v}(v)$$



product of incoming messages from all other neighbours

factor to variable messages

$$\mu_{f\to v}(v) = \sum_{\mathcal{X}_f\setminus v} f(\mathcal{X}_f) \prod_{x\in\mathcal{X}_f\setminus v} \mu_{x\to f}(x)$$



sum over all values of all variables in f except v, of
product of incoming messages from these variables times the factor

Sum-Product Algorithm

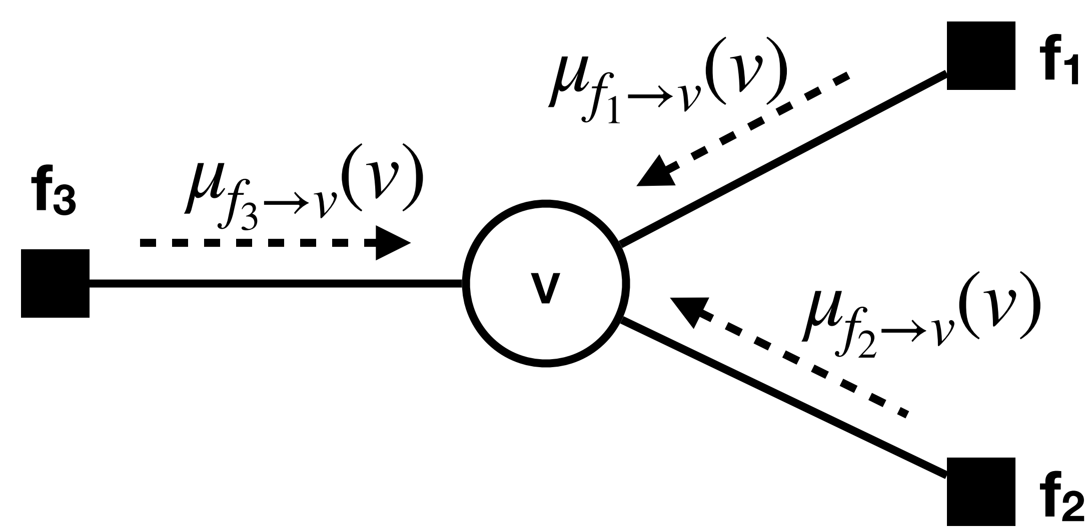
Marginal of a variable v

is proportional to the product of incoming messages from all neighbours

$$P(v) \propto \prod_{f_i \in ne(v)} \mu_{f_i \to v}(v)$$

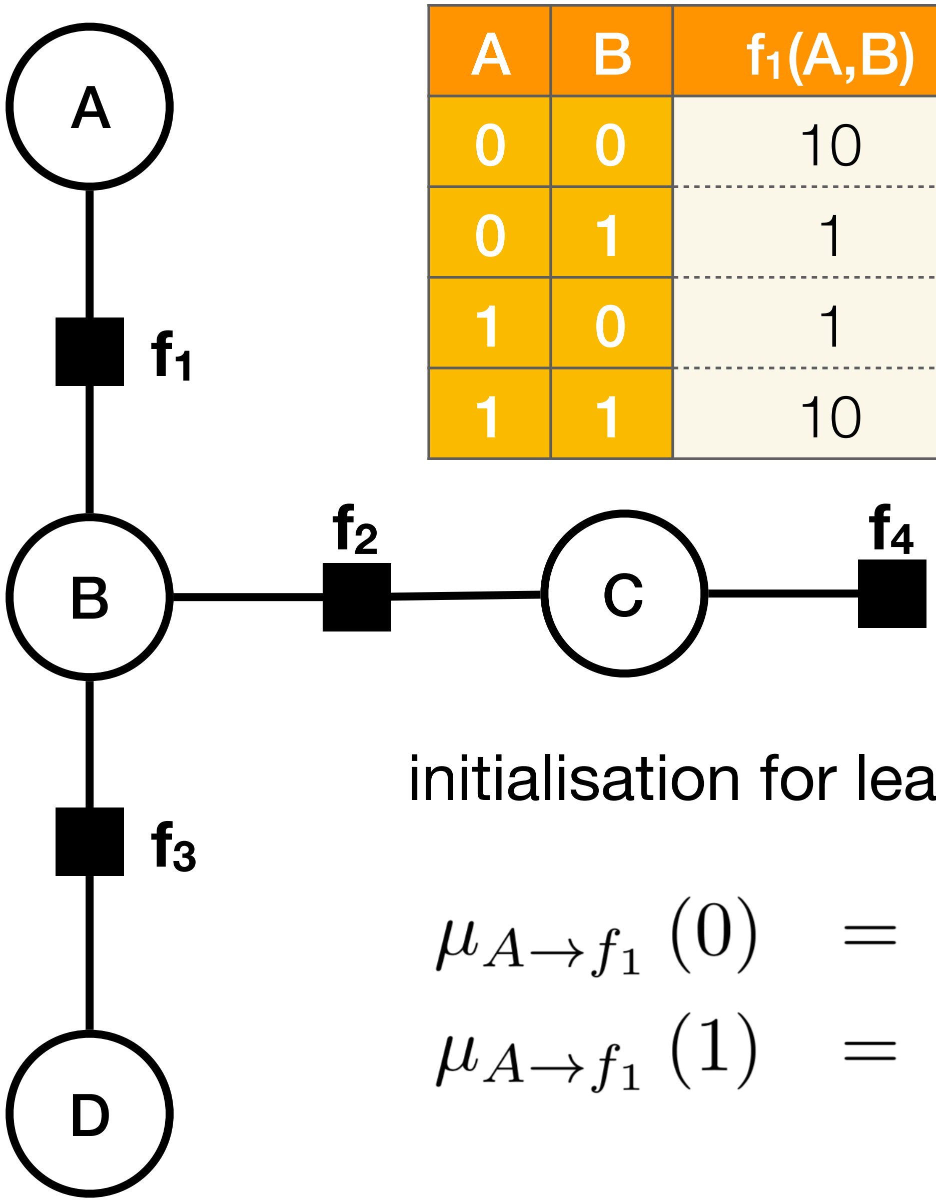
to normalise, use fact that

$$\sum_{v \in dom(V)} P(V = v) \text{ equals } 1$$



Example

$$P(A,B,C,D) \propto f_1(A,B) \cdot f_2(B,C) \cdot f_3(B,D) \cdot f_4(C)$$



A	B	$f_1(A,B)$
0	0	10
0	1	1
1	0	1
1	1	10

B	C	$f_2(B,C)$
0	0	1
0	1	10
1	0	10
1	1	1

B	D	$f_3(B,D)$
0	0	10
0	1	1
1	0	1
1	1	10

C	$f_4(C)$
0	10
1	1

pick a root: f_4

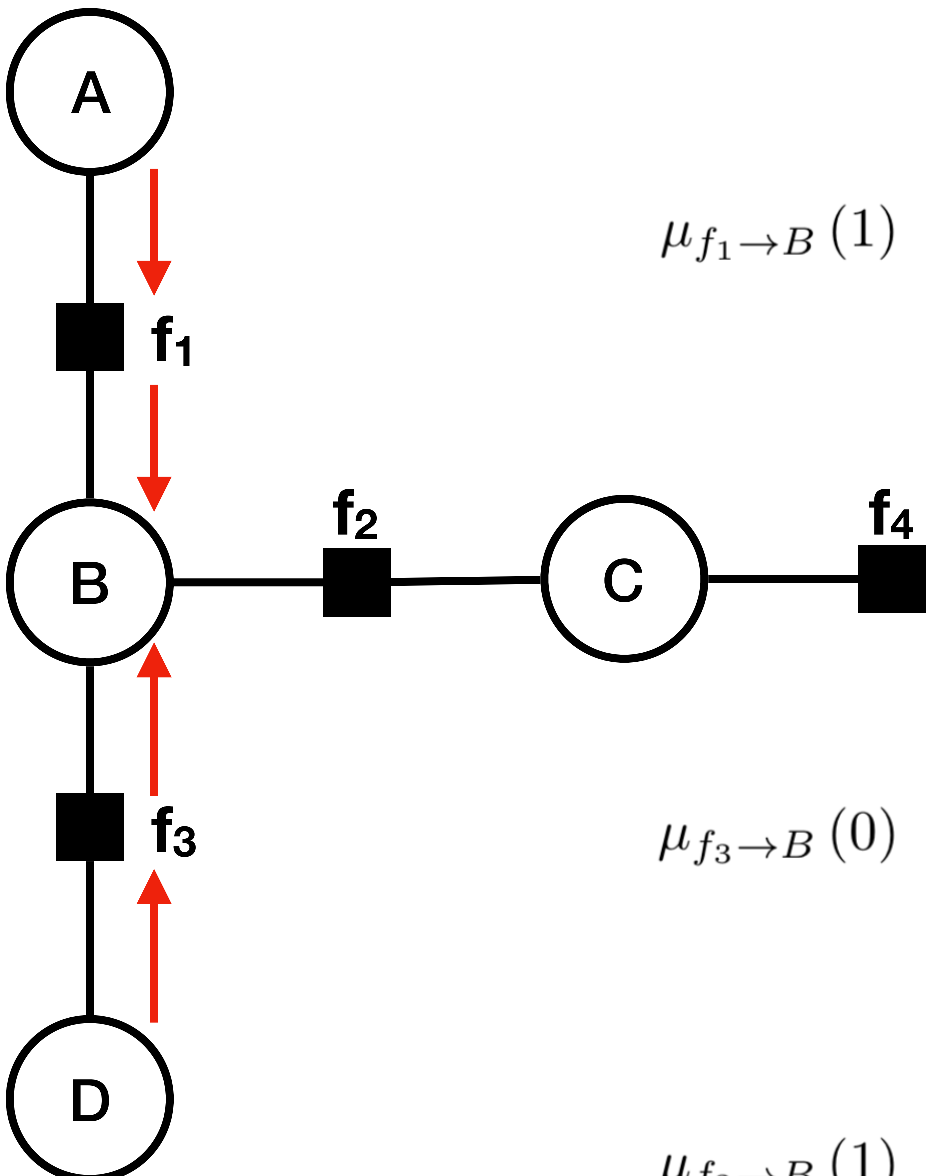
initialisation for leaves:

$$\mu_{A\to f_1}(0) = 1$$
$$\mu_{A\to f_1}(1) = 1$$

$$\mu_{f_4\to C}(0) = f_4(C=0) = 10$$
$$\mu_{f_4\to C}(1) = f_4(C=1) = 1$$

$$\mu_{D\to f_3}(0) = 1$$
$$\mu_{D\to f_3}(1) = 1$$

now, propagate towards root f_4, starting from A and D



f_1 has all messages needed to propagate to B:

$$\begin{aligned}
\mu_{f_1\to B}(0) &= \sum_A f_1(A, B=0)\,\mu_{A\to f_1}(A)\\
&= \underbrace{f_1(A=0,B=0)}_{10}\underbrace{\mu_{A\to f_1}(0)}_{1} + \underbrace{f_1(A=1,B=0)}_{1}\underbrace{\mu_{A\to f_1}(1)}_{1} = 11\\
\mu_{f_1\to B}(1) &= \sum_A f_1(A, B=1)\,\mu_{A\to f_1}(A)\\
&= \underbrace{f_1(A=0,B=1)}_{1}\underbrace{\mu_{A\to f_1}(0)}_{1} + \underbrace{f_1(A=1,B=1)}_{10}\underbrace{\mu_{A\to f_1}(1)}_{1} = 11
\end{aligned}$$

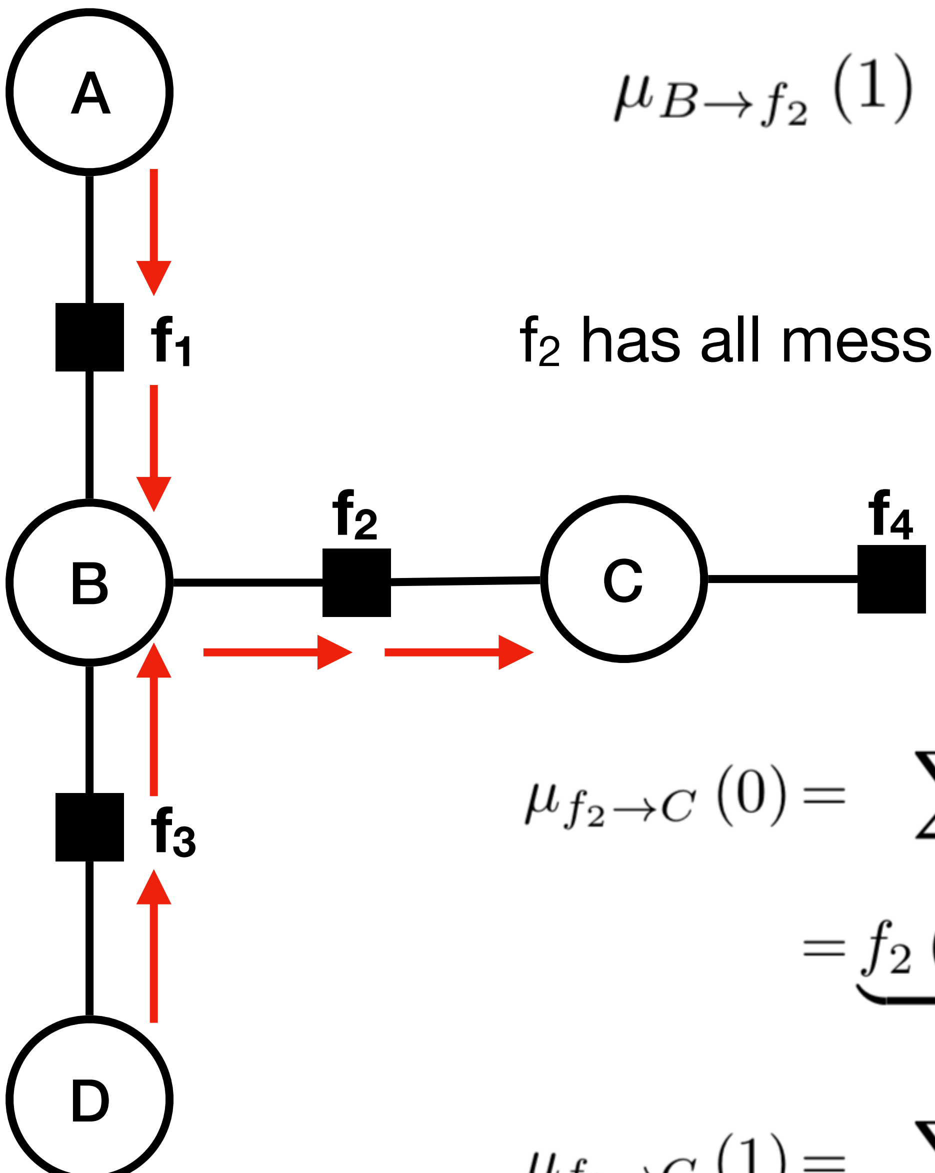
f_3 has all messages needed to propagate to B:

$$\begin{aligned}
\mu_{f_3\to B}(0) &= \sum_D f_3(B=0, D)\,\mu_{D\to f_3}(D)\\
&= \underbrace{f_3(B=0,D=0)}_{10}\underbrace{\mu_{D\to f_3}(0)}_{1} + \underbrace{f_3(B=0,D=1)}_{1}\underbrace{\mu_{D\to f_3}(1)}_{1} = 11\\
\mu_{f_3\to B}(1) &= \sum_D f_3(B=1, D)\,\mu_{D\to f_3}(D)\\
&= \underbrace{f_3(B=1,D=0)}_{1}\underbrace{\mu_{D\to f_3}(0)}_{1} + \underbrace{f_3(B=1,D=1)}_{10}\underbrace{\mu_{D\to f_3}(1)}_{1} = 11
\end{aligned}$$

B has all messages needed to propagate to f_2:

$$\mu_{B\to f_2}(0) = \underbrace{\mu_{f_1\to B}(0)}_{11}\underbrace{\mu_{f_3\to B}(0)}_{11} = 121$$

$$\mu_{B\to f_2}(1) = \underbrace{\mu_{f_1\to B}(1)}_{11}\underbrace{\mu_{f_3\to B}(1)}_{11} = 121$$



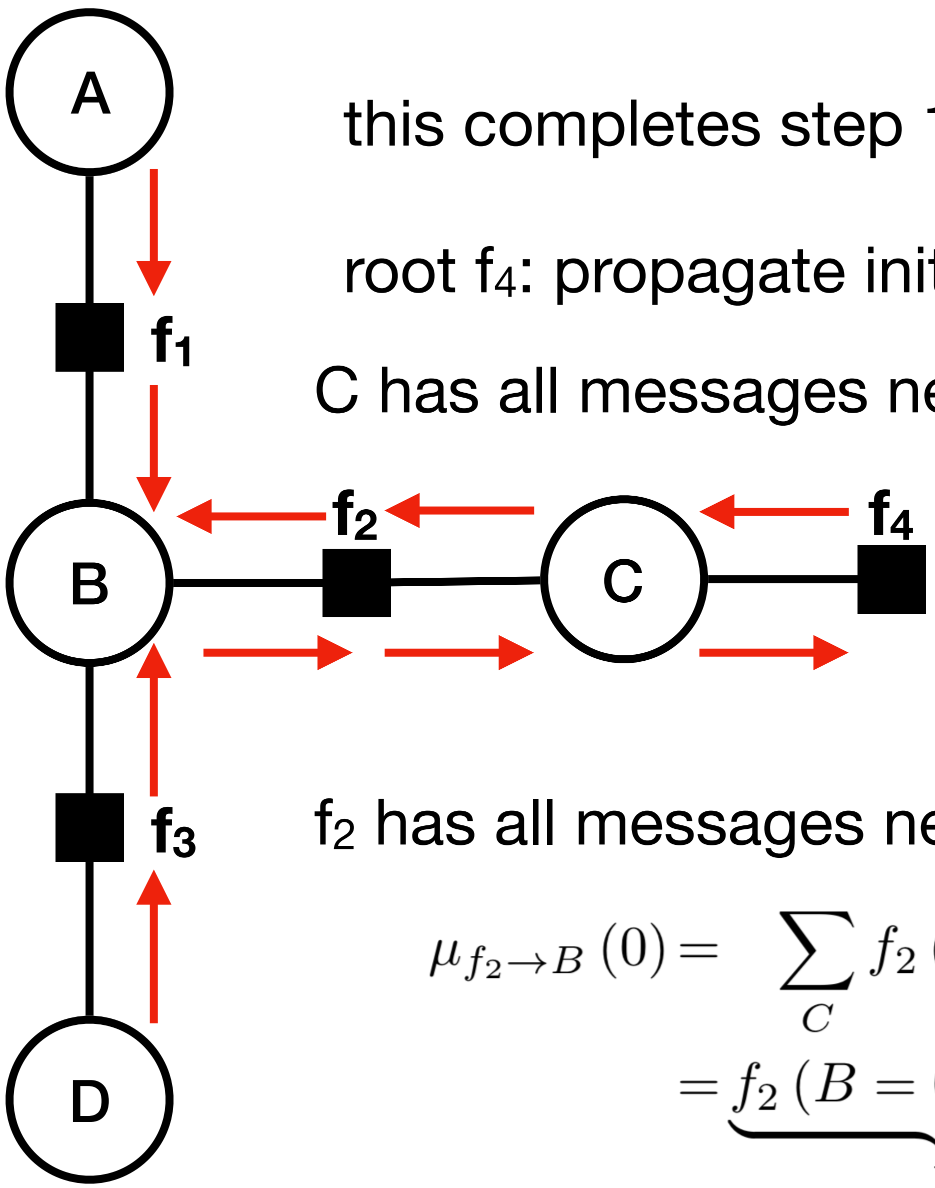
f_2 has all messages needed to propagate to C:

$$\begin{aligned}\mu_{f_2\to C}(0) &= \sum_B f_2(B, C=0)\,\mu_{B\to f_2}(B)\\ &= \underbrace{f_2(B=0, C=0)}_{1}\underbrace{\mu_{B\to f_2}(0)}_{121} + \underbrace{f_2(B=1, C=0)}_{10}\underbrace{\mu_{B\to f_2}(1)}_{121} = 1331\end{aligned}$$

$$\begin{aligned}\mu_{f_2\to C}(1) &= \sum_B f_2(B, C=1)\,\mu_{B\to f_2}(B)\\ &= \underbrace{f_2(B=0, C=1)}_{10}\underbrace{\mu_{B\to f_2}(0)}_{121} + \underbrace{f_2(B=1, C=1)}_{1}\underbrace{\mu_{B\to f_2}(1)}_{121} = 1331\end{aligned}$$

C has all messages needed to propagate to f_4:

$$\begin{aligned}\mu_{C\to f_4}(0) &= \mu_{f_2\to C}(0) = 1331\\ \mu_{C\to f_4}(1) &= \mu_{f_2\to C}(1) = 1331\end{aligned}$$



this completes step 1; step 2 propagates from the root back to the leaves

root f_4: propagate initialised message

C has all messages needed to propagate to f_2:

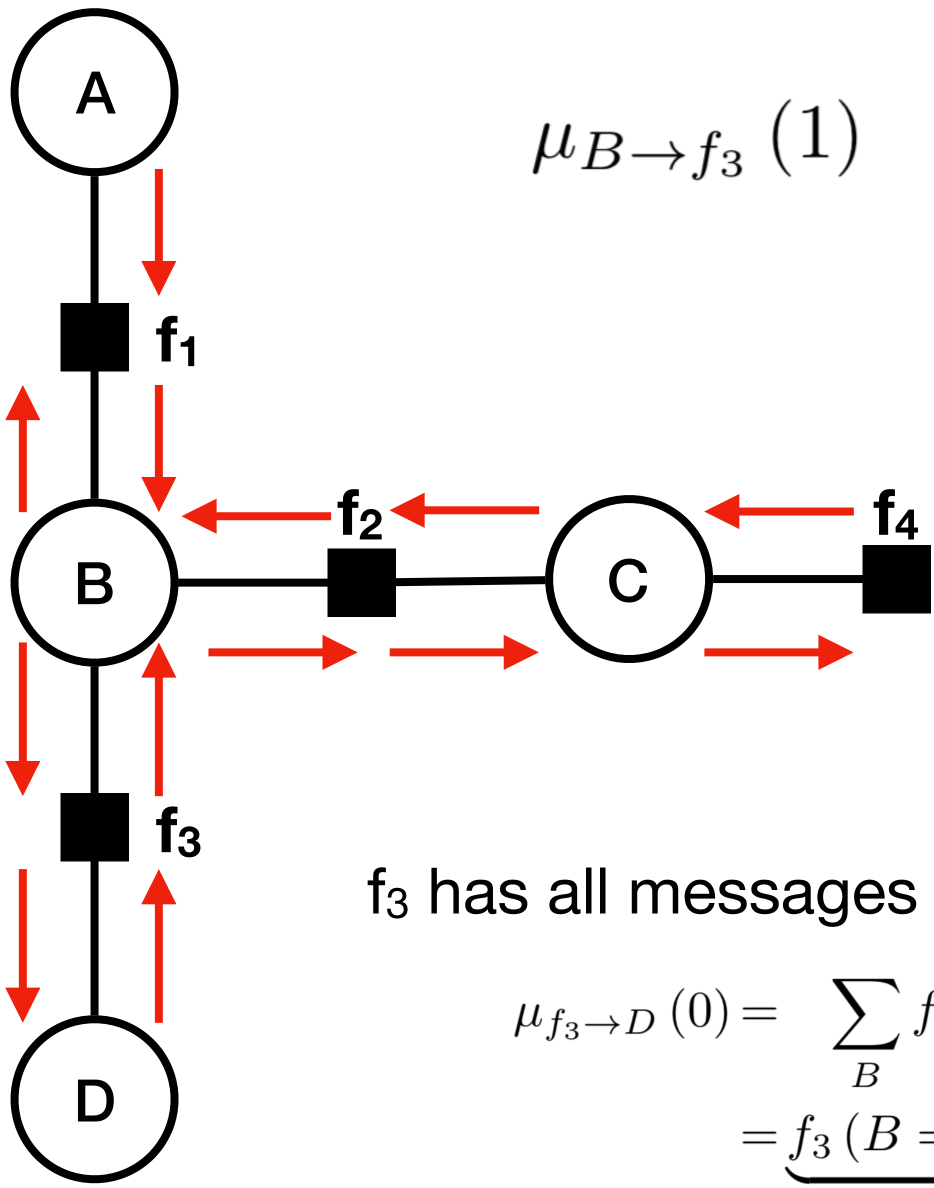
$$\begin{aligned}\mu_{C\to f_2}(0) &= \mu_{f_4\to C=0}(C=0) = 10\\ \mu_{C\to f_2}(1) &= \mu_{f_4\to C=1}(C=1) = 1\end{aligned}$$

f_2 has all messages needed to propagate to B:

$$\begin{aligned}\mu_{f_2\to B}(0) &= \sum_C f_2(B=0,C)\,\mu_{C\to f_2}(C)\\ &= \underbrace{f_2(B=0,C=0)}_{1}\underbrace{\mu_{C\to f_2}(0)}_{10} + \underbrace{f_2(B=0,C=1)}_{10}\underbrace{\mu_{C\to f_2}(1)}_{1} = 20\end{aligned}$$

$$\begin{aligned}\mu_{f_2\to B}(1) &= \sum_C f_2(B=1,C)\,\mu_{C\to f_2}(C)\\ &= \underbrace{f_2(B=1,C=0)}_{10}\underbrace{\mu_{C\to f_2}(0)}_{10} + \underbrace{f_2(B=1,C=1)}_{1}\underbrace{\mu_{C\to f_2}(1)}_{1} = 101\end{aligned}$$

B has all messages needed to propagate to f_3 and to f_1:



$$\mu_{B\to f_3}(0) = \underbrace{\mu_{f_1\to B}(0)}_{11}\underbrace{\mu_{f_2\to B}(0)}_{20} = 220$$

$$\mu_{B\to f_3}(1) = \underbrace{\mu_{f_1\to B}(1)}_{11}\underbrace{\mu_{f_2\to B}(1)}_{101} = 1111$$

$$\mu_{B\to f_1}(0) = \underbrace{\mu_{f_2\to B}(0)}_{20}\underbrace{\mu_{f_3\to B}(0)}_{11} = 220$$

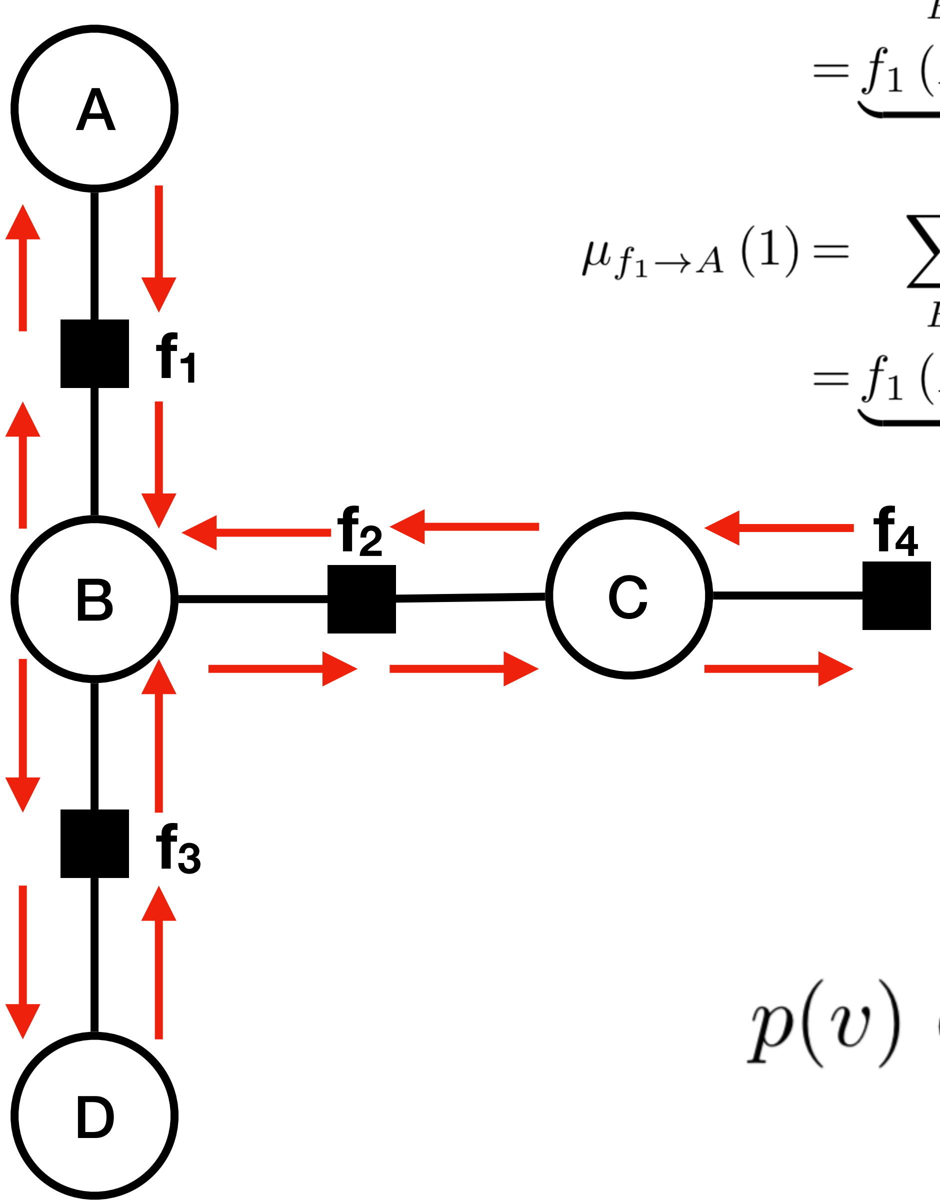
$$\mu_{B\to f_1}(1) = \underbrace{\mu_{f_2\to B}(1)}_{101}\underbrace{\mu_{f_3\to B}(1)}_{11} = 1111$$

f_3 has all messages needed to propagate to D:

$$\begin{aligned}\mu_{f_3\to D}(0) &= \sum_B f_3(B, D=0)\,\mu_{B\to f_3}(B)\\ &= \underbrace{f_3(B=0, D=0)}_{10}\underbrace{\mu_{B\to f_3}(0)}_{220} + \underbrace{f_3(B=1, D=0)}_{1}\underbrace{\mu_{B\to f_3}(1)}_{1111} = 3311\end{aligned}$$

$$\begin{aligned}\mu_{f_3\to D}(1) &= \sum_B f_3(B, D=1)\,\mu_{B\to f_3}(B)\\ &= \underbrace{f_3(B=0, D=1)}_{1}\underbrace{\mu_{B\to f_3}(0)}_{220} + \underbrace{f_3(B=1, D=1)}_{10}\underbrace{\mu_{B\to f_3}(1)}_{1111} = 11330\end{aligned}$$

f_1 has all messages needed to propagate to A:



$$\mu_{f_1\to A}(0) = \sum_B f_1(A=0,B)\,\mu_{B\to f_1}(B)$$
$$= \underbrace{f_1(A=0,B=0)}_{10}\underbrace{\mu_{B\to f_1}(0)}_{220} + \underbrace{f_1(A=0,B=1)}_{1}\underbrace{\mu_{B\to f_1}(1)}_{1111} = 3311$$

$$\mu_{f_1\to A}(1) = \sum_B f_1(A=1,B)\,\mu_{B\to f_1}(B)$$
$$= \underbrace{f_1(A=1,B=0)}_{1}\underbrace{\mu_{B\to f_1}(0)}_{220} + \underbrace{f_1(A=1,B=1)}_{10}\underbrace{\mu_{B\to f_1}(1)}_{1111} = 11330$$

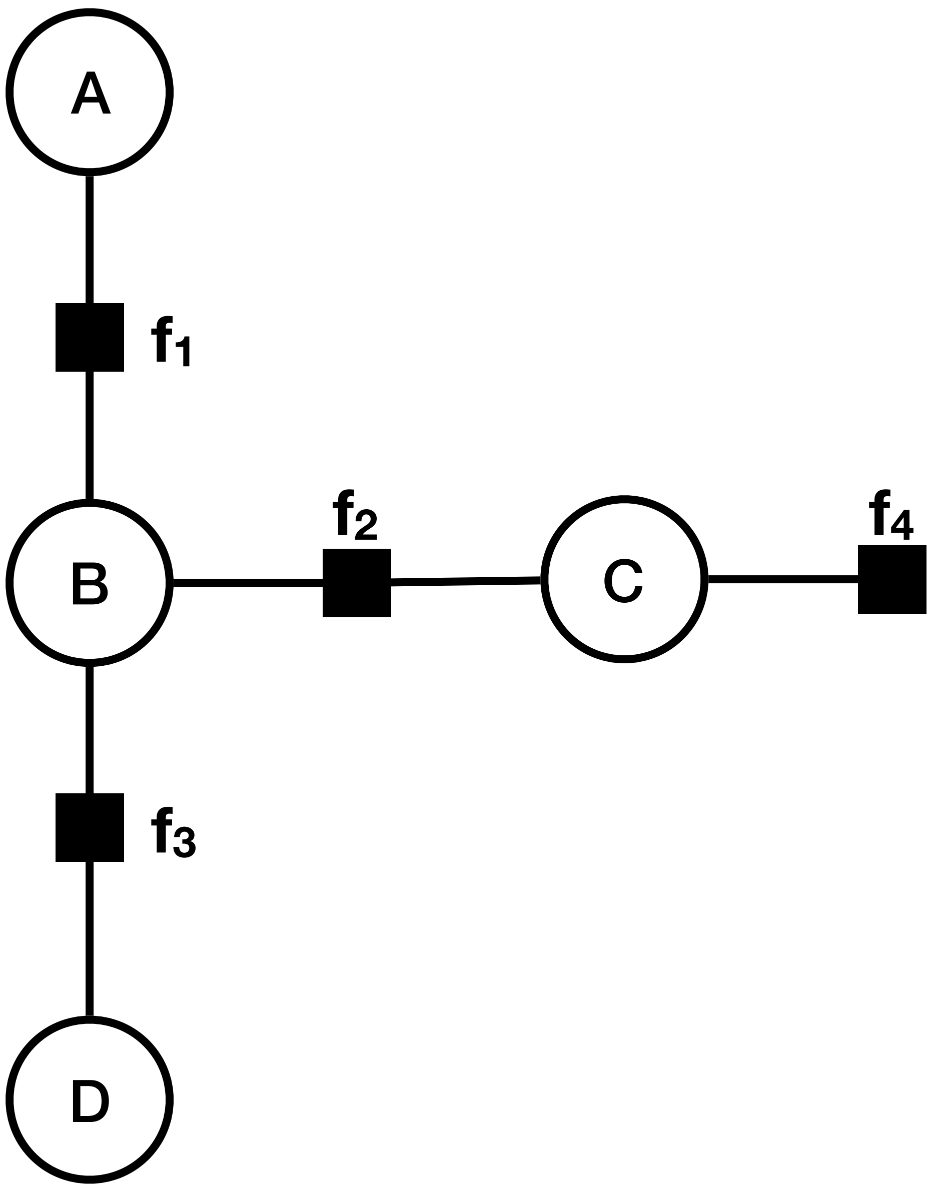
done!

now we can use the messages to compute marginals:

$$p(v) \propto \prod\nolimits_{f_i\sim v} \mu_{f_i\to v}(v)$$

$$Z = \sum_{\mathcal{X}} \prod_f \phi_f(\mathcal{X}_f) \text{ as } Z = \sum_x \prod_{f\in ne(x)} \mu_{f\to x}(x)$$

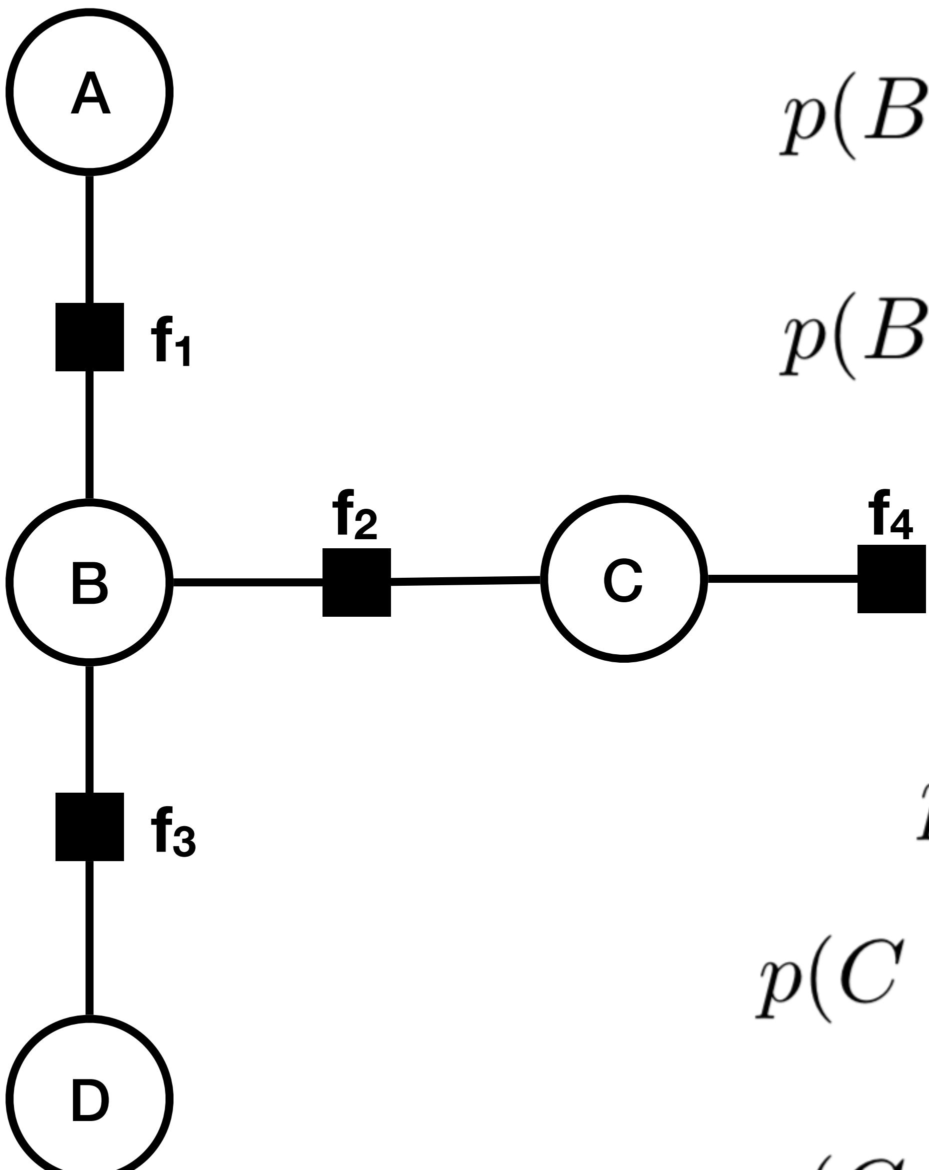
computing marginals using $p(v) \propto \prod_{f_i \sim v} \mu_{f_i \to v}(v)$



$$\begin{aligned} p(A) &\propto \mu_{f_1 \to A}(A) \\ p(A=0) &= \frac{3311}{3311 + 11330} = 0.23 \\ p(A=1) &= \frac{11330}{3311 + 11330} = 0.77 \end{aligned}$$

$$\begin{aligned} p(D) &\propto \mu_{f_3 \to D}(D) \\ p(D=0) &= \frac{3311}{3311 + 11330} = 0.23 \\ p(D=1) &= \frac{11330}{3311 + 11330} = 0.77 \end{aligned}$$

computing marginals using $p(v) \propto \prod_{f_i \sim v} \mu_{f_i \to v}(v)$



$$p(B) \propto \mu_{f_1 \to B}(B)\, \mu_{f_2 \to B}(B)\, \mu_{f_3 \to B}(B)$$

$$p(B=0) = \frac{11 * 20 * 11}{11 * 20 * 11 + 11 * 101 * 11} = 0.17$$

$$p(B=1) = \frac{11 * 101 * 11}{11 * 20 * 11 + 11 * 101 * 11} = 0.83$$

$$p(C) \propto \mu_{f_2 \to C}(C)\, \mu_{f_4 \to C}(C)$$

$$p(C=0) = \frac{1331 * 10}{1331 * 10 + 1331 * 1} = 0.91$$

$$p(C=1) = \frac{1331 * 1}{1331 * 10 + 1331 * 1} = 0.09$$

Sum-product algorithm

- FGs can have loops, which cause a problem for message passing: eliminating a variable may introduce a factor that isn't in the graph yet
- what to do?
  - option 1: loopy belief propagation
    - use propagation rules anyways, hoping that messages will converge
    - no guarantees, but often works well enough in practice
  - option 2: bucket elimination
    - guaranteed to produce correct answers

Bucket Elimination

- A variable elimination / message passing approach that computes the marginal of a variable X on any FG (with or without loops)
- main steps:
  1. choose variable order starting with X
  2. distribute potentials over buckets
  3. iteratively eliminate buckets until only one bucket left
- we'll trace the algorithm using a table
  - rows correspond to buckets (ordered bottom-up)
  - columns are iterations

Bucket Elimination

- fix variable order X1,X2,…,Xn
- distributing potentials over buckets:
  - for i=1,…,n
    - add all potentials involving Xi that are not yet in any bucket to bucket i
- eliminating buckets:
  - for i=1,…,n-1
    - marginalise the product of the entries in bucket i over Xi
    - add the result to bucket j, where Xj is the first variable in the order present in the result

Example $P(F) = \sum_{a,b,c,d,e,g} P(F\,|\,d)P(g\,|\,d,e)P(c\,|\,a)P(d\,|\,a,b)P(a)P(b)P(e)$

variable order F,D,A,G,B,C,E

E	P(G|D,E), P(E)						
C	P(C|A)	P(C|A)					
B	P(D|A,B), P(B)	P(D|A,B), P(B)	P(D|A,B), P(B)				
G		f_E(D,G)	f_E(D,G)	f_E(D,G)			
A	P(A)	P(A)	P(A)	P(A), f_B(D,A)	P(A), f_B(D,A)		
D	P(F|D)	P(F|D)	P(F|D)	P(F|D)	P(F|D), f_G(D)	P(F|D), f_G(D), f_A(D)	
F							f_D(F)

$$f_E(D,G) = \sum_E P(G\,|\,D,E)P(E)$$

$$f_C(A) = \sum_C P(C\,|\,A) = 1$$

$$f_B(D,A) = \sum_B P(D\,|\,B,D)P(B)$$

$$f_G(D) = \sum_G f_E(D,G)$$

$$f_A(D) = \sum_A P(A) f_B(D,A)$$

$$f_D(F) = \sum_D P(F\,|\,D) f_G(D) f_A(D)$$

Bucket Elimination

- computes the marginal of one variable only
- multi-variable messages need storage exponential in number of their variables
- for graphs without loops, computational complexity depends on ordering: there is an order with linear complexity, but others are much worse

Today

- **Graphical Models**:
  - **Markov networks** as alternative representation of factored distributions
  - **Factor graphs** as basis for inference
- **Reasoning** with Graphical Models
  - **Variable Elimination** on Bayesian networks
  - **Sum-product algorithm** on factor graphs
  - **Bucket elimination** as alternative for loopy graphs

Reading Material

- Today:
  - Russell & Norvig: 14.4
  - Barber: chapters 4 & 5
- Next week:
  - Russell & Norvig: 14.5
  - Barber: 6 & 27 (yes, that’s 27)

- Parts of slides based on
  - David Barber’s slides for the BRML book
  - Tinne De Laet & Luc De Raedt’s slides for the UAI course at KU Leuven