ECE521 W17 Tutorial 8

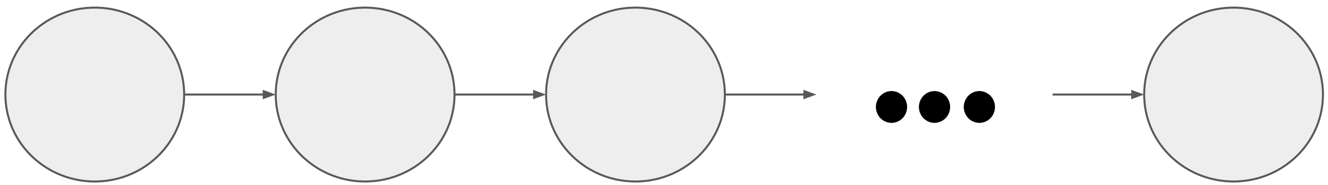
Eleni Triantafillou and Yuhuai (Tony) Wu

Some slides borrowed from last year's tutorial, Eric Xing's course and some figures from Bishop's book and others

UNIVERSITY OF TORONTO

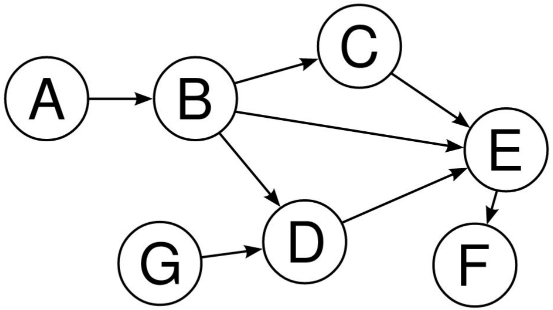
Conditional Independence

- We are often interested in computing joint probability distributions
- It is desirable to decompose it into a product of factors, each depending on a subset of the variables, for ease of computation.
- Conditional independence properties between the variables allow us to do this.
- A common example of conditional independence: Markov chains.
  We assume that the future is independent of the past given the present.



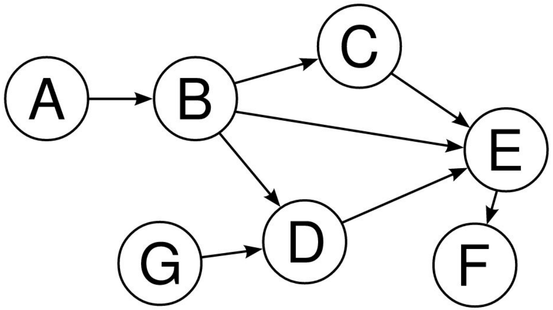
Graphical models

- Bayesian networks (i.e. BN, BayesNet), directed-acyclic-graph (DAG)

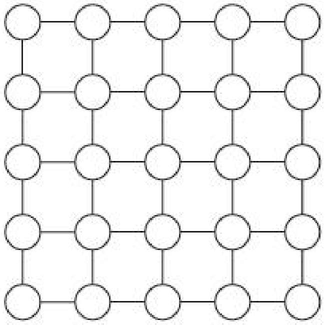


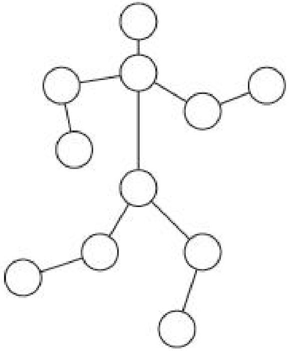
Graphical models

- Bayesian networks (i.e. BN, BayesNet), directed-acyclic-graph (DAG)



- Markov random fields, undirected graph





Bayesian Network:

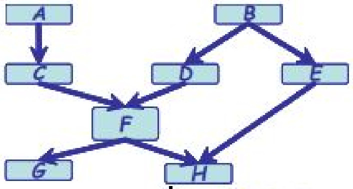
- A BN is a directed graph whose nodes represent the random variables and whose edges represent direct influence of one variable on another.

- It is a data structure that provides the skeleton for representing **a joint distribution** compactly in a **factorized** way;

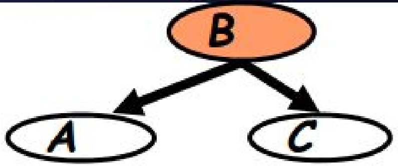
- It offers a compact representation for **a set of conditional independence assumptions** about a distribution;

- We can view the graph as encoding a generative sampling process executed by nature, where the value for each variable is selected by nature using a distribution that depends only on its parents. In other words, each variable is a stochastic function of its parents.

Local Structures & Independencies

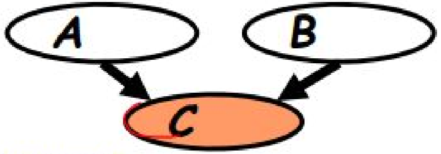


- Common parent
  - Fixing B decouples A and C
    "given the level of gene B, the levels of A and C are independent"

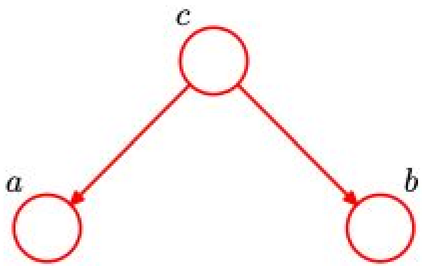


- Cascade
  - Knowing B decouples A and C
    "given the level of gene B, the level gene A provides no extra prediction value for the level of gene C"

- V-structure
  - Knowing C couples A and B
    because A can "explain away" B w.r.t. C
    "If A correlates to C, then chance for B to also correlate to B will decrease"



Common parent



According to the graphical model, we can decompose the joint probability over the 3 variables as:

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

In general, we have: $$p(a, b) = \sum_c p(a|c)p(b|c)p(c)$$

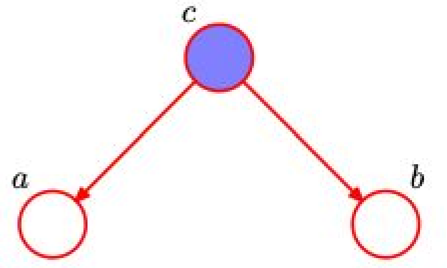
This does not in general decompose into: $$p(a, b) = p(a)p(b)$$

So a and b are not independent.

Common parent

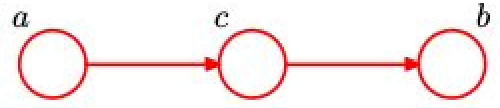
… But if we observe c:

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



So a and b are **conditionally** independent given c

Cascade



According to the graphical model we can decompose the joint as:

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

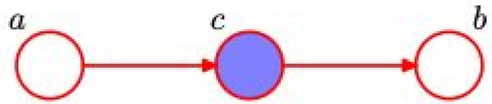
In general, we have:

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

Which does not in general factorize as: $p(a, b) = p(a)p(b)$

So a and b are not independent

Cascade

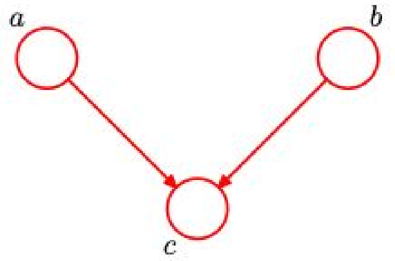


But if we condition on c...

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

a and b are **conditionally independent** given c

V-structure



According to the graphical model we can decompose the joint as:

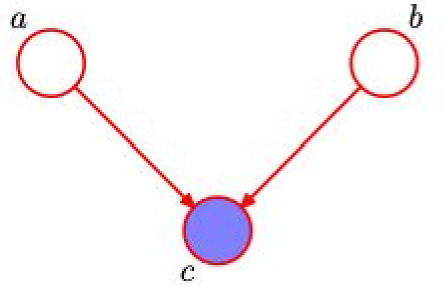
$$p(a,b,c) = p(a)p(b)p(c|a,b)$$

In general, we have:

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

So a and b are independent!

V-structure



… but if we condition on c:

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

which does does not in general factorize into $p(a)p(b)$

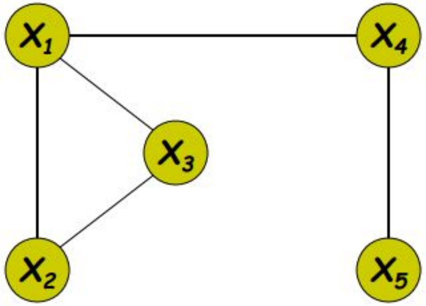
Therefore a and b are not conditionally independent given c.

Active trail

- **Causal trail** X → Z → Y **:** active if and only if Z is not observed.
- **Evidential trail** X ← Z ← Y **:** active if and only if Z is not observed.
- **Common cause** X ← Z → Y **:** active if and only if Z is not observed.
- **Common effect** X → Z ← Y **:** active if and only if either Z or one of Z's descendants is observed

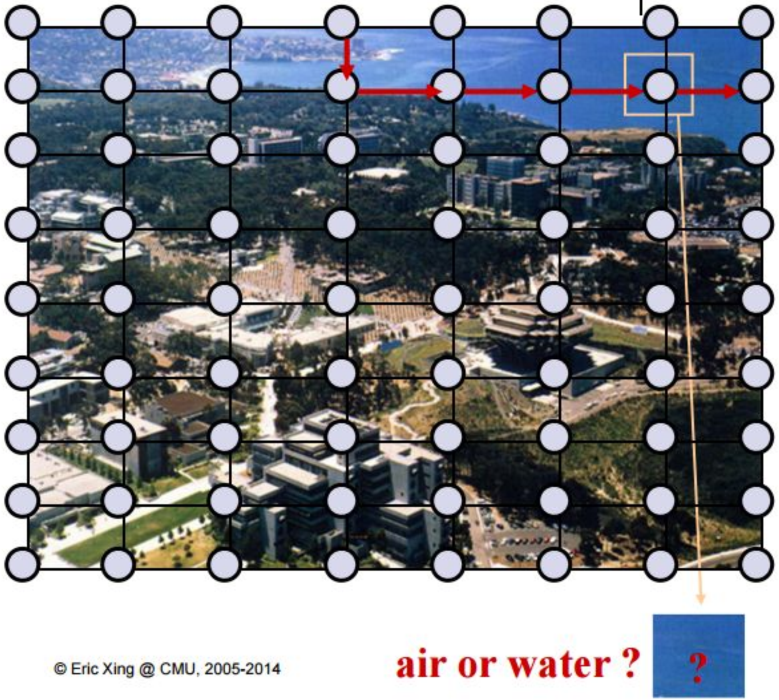
Definition : Let $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ be three sets of nodes in *G*. We say that $\mathbf{X}$ and $\mathbf{Y}$ are *d-separated given* $\mathbf{Z}$, denoted *d-sep*$_{\mathcal{G}}(\mathbf{X};\mathbf{Y} \mid \mathbf{Z})$, if there is no active trail between any node $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ given $\mathbf{Z}$.

Undirected graphical models (UGM)



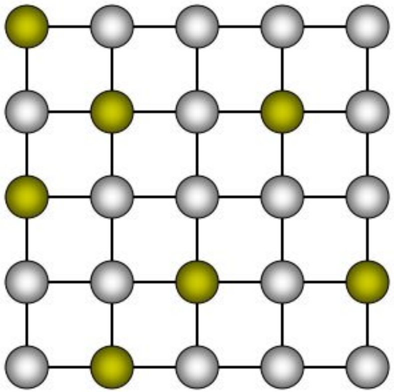
- Pairwise (non-causal) relationships
- Can write down model, and score specific configurations of the graph, but no explicit way to generate samples
- Contingency constrains on node configurations

A Canonical Examples: understanding complex scene ...



Canonical example

- The grid model



- Naturally arises in image processing, lattice physics, etc.
- Each node may represent a single "pixel", or an atom
  - The states of adjacent or nearby nodes are "coupled" due to pattern continuity or electro-magnetic force, etc.
  - Most likely joint-configurations usually correspond to a "low-energy" state

Representation

- Defn: an undirected graphical model represents a distribution $P(X_1, \ldots, X_n)$ defined by an undirected graph H, and a set of positive ***potential functions*** y_c associated with the cliques of H, s.t.

$$P(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{c \in C} \psi_c(\mathbf{x}_c)$$

where Z is known as the partition function:

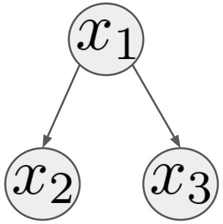
$$Z = \sum_{x_1, \ldots, x_n} \prod_{c \in C} \psi_c(\mathbf{x}_c)$$

- Also known as Markov Random Fields, Markov networks …
- The ***potential function*** can be understood as an contingency function of its arguments assigning "pre-probabilistic" score of their joint configuration.

Factor Graphs

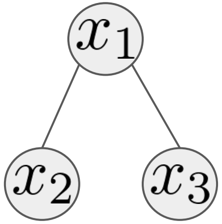
- Both directed and undirected graphical models express a joint probability distribution in a factorized way. For example:
- Directed:

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



- Undirected:

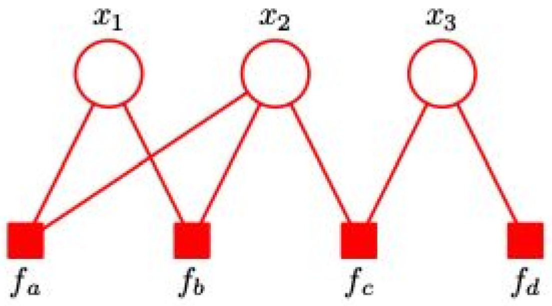
$$p(x_1, x_2, x_3) = \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)$$



Factor Graphs

Let us write the joint distribution over a set of variables in the form of a product of factors (with x_s denoting a subset of variables):

$$p(x) = \prod_s f_s(x_s)$$



Factor graphs have nodes for variables as before (circles) and also for factors (squares). This can be used to represent either a directed or undirected PGM.

Example factor graphs for directed GM

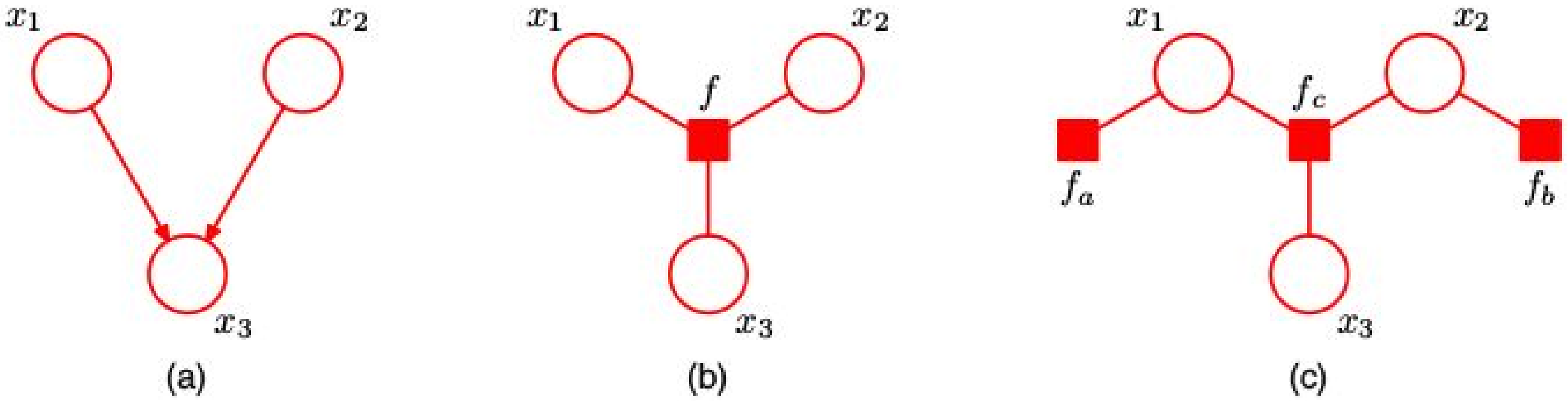


Figure 8.42 (a) A directed graph with the factorization $p(x_1)p(x_2)p(x_3|x_1, x_2)$. (b) A factor graph representing the same distribution as the directed graph, whose factor satisfies $f(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2)$. (c) A different factor graph representing the same distribution with factors $f_a(x_1) = p(x_1)$, $f_b(x_2) = p(x_2)$ and $f_c(x_1, x_2, x_3) = p(x_3|x_1, x_2)$.

Example factor graphs for undirected GM

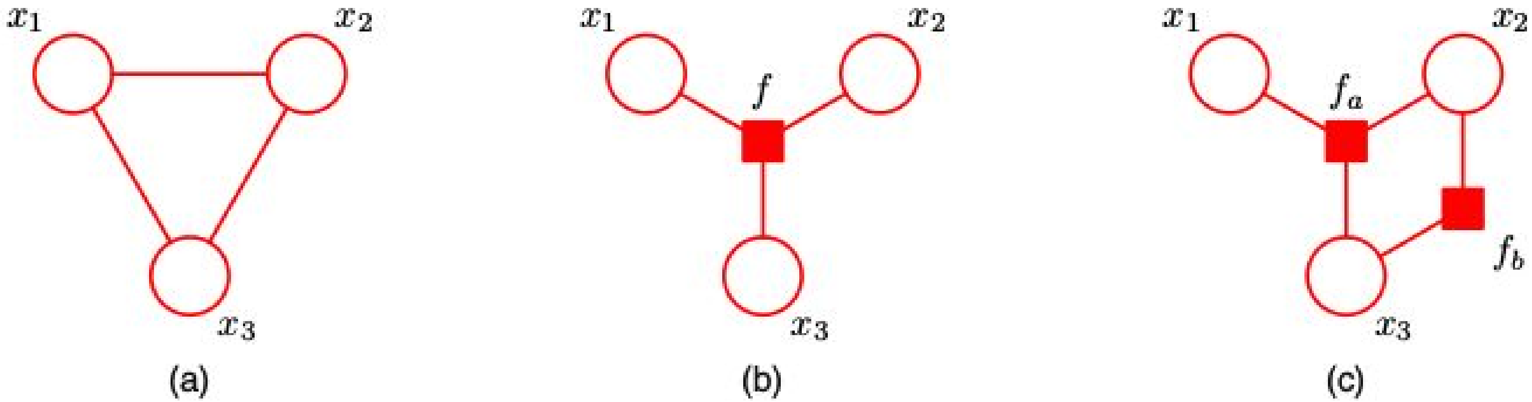


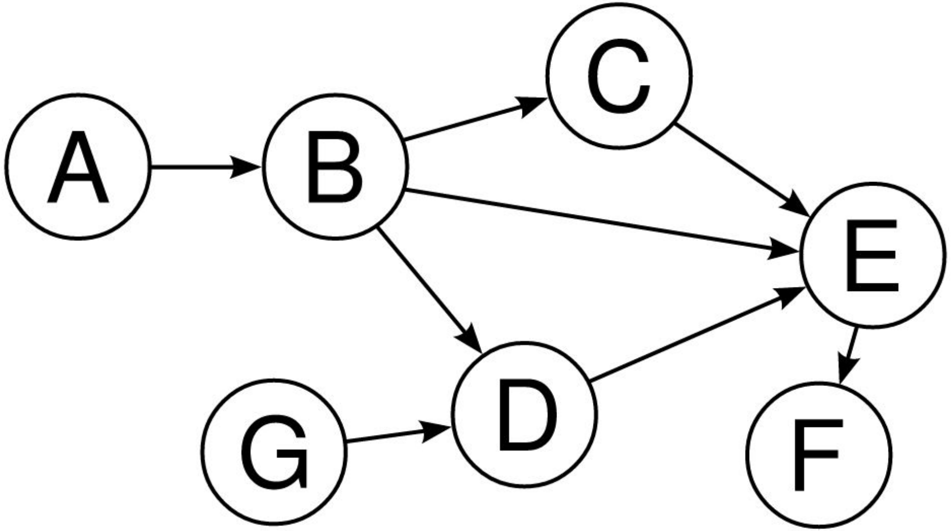
Figure 8.41 (a) An undirected graph with a single clique potential $\psi(x_1, x_2, x_3)$. (b) A factor graph with factor $f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$ representing the same distribution as the undirected graph. (c) A different factor graph representing the same distribution, whose factors satisfy $f_a(x_1, x_2, x_3) f_b(x_1, x_2) = \psi(x_1, x_2, x_3)$.

Conditional independence in factor graph

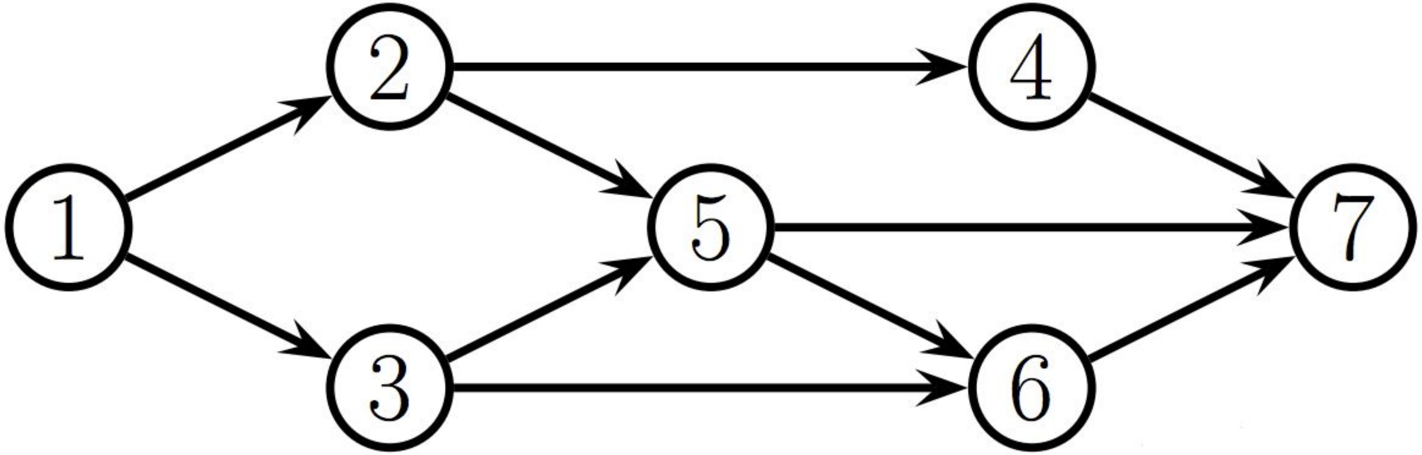
- The Markov blanket for our factor graphs is very similar to MRFs

- The Markov blanket of a variable node in a factor graph is given by the variables' **second neighbours**

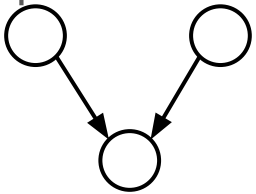
Conditional independence in Bayesian nets examples



Conditional independence in Bayesian nets examples



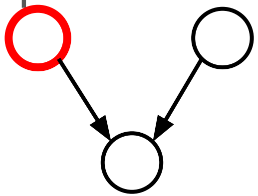
BNs ⟺ factor graph



- Converting Bayesian Networks to factor graph takes the following steps:
    - Consider all the parents of a child node
    - “Pinch” all the edges from its parents to the child into one factor
    - Create an additional edge from the factor to the child node
    - Move on the the next child node
    - Last step is to add all the priors as individual “dongles” to the corresponding variables
- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

BNs ⟺ factor graph

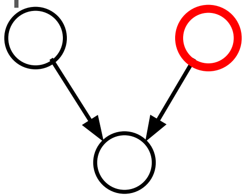
- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables



- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

BNs ⟺ factor graph

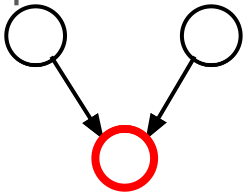
- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables



- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

BNs ⟺ factor graph

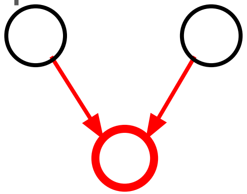
- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables



- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

BNs ⟺ factor graph

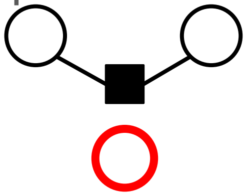
- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables



- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

BNs ⟺ factor graph

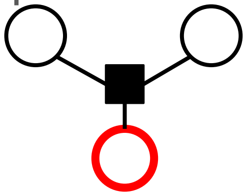
- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables



- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

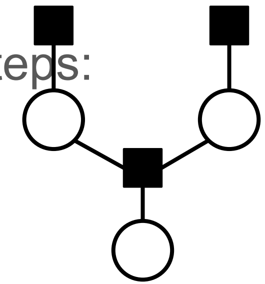
BNs ⟺ factor graph

- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables



- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

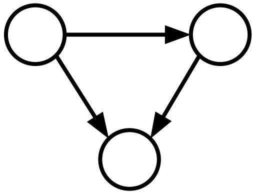
BNs ⟺ factor graph

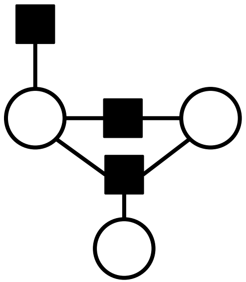


- Converting Bayesian Networks to factor graph takes the following steps:
    - Consider all the parents of a child node
    - “Pinch” all the edges from its parents to the child into one factor
    - Create an additional edge from the factor to the child node
    - Move on the the next child node
    - Last step is to add all the priors as individual “dongles” to the corresponding variables
- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

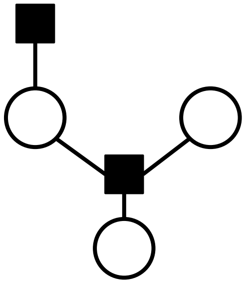
BNs ⟺ factor graph

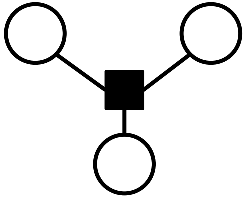
- With this approach you may get factor graphs like the following:





which can be simplified to:

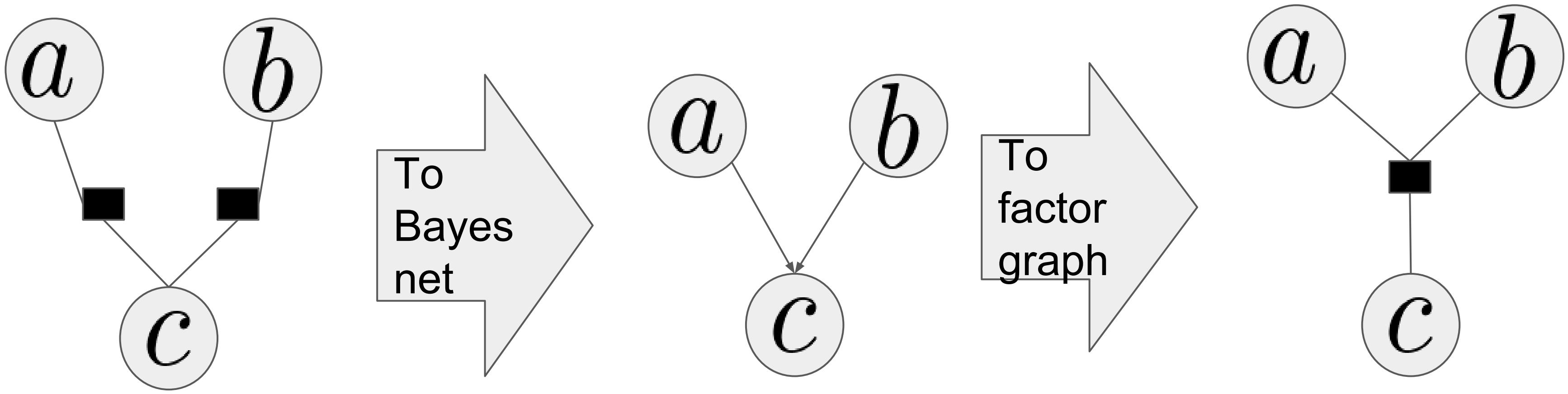




BNs ⟺ factor graph

- Convert FG back to BN by just reserving the “pinching” on each factor node
- Then put back the direction on the edge according to the conditional probabilities

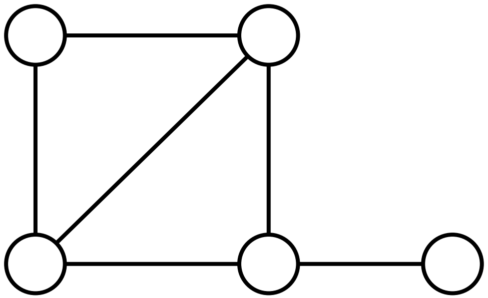
BNs $\Longleftrightarrow$ factor graph



Notice that we don't get the same factor graph back...

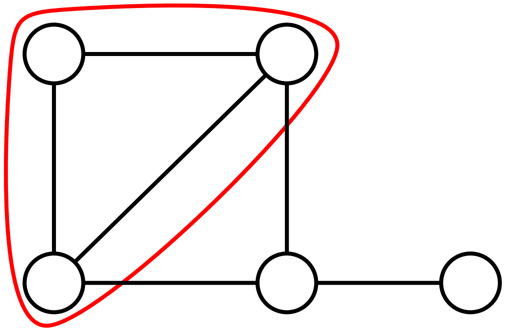
MRF ⟺ factor graph

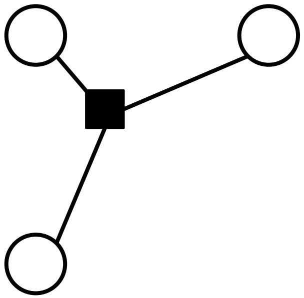
- Converting Markov Random Fields to factor graph takes the following steps:
  - Consider all the maximum cliques of the MRF
  - Create a factor node for each of the maximum cliques
  - Connect all the nodes of the maximum clique to the new factor nodes



MRF ⟺ factor graph

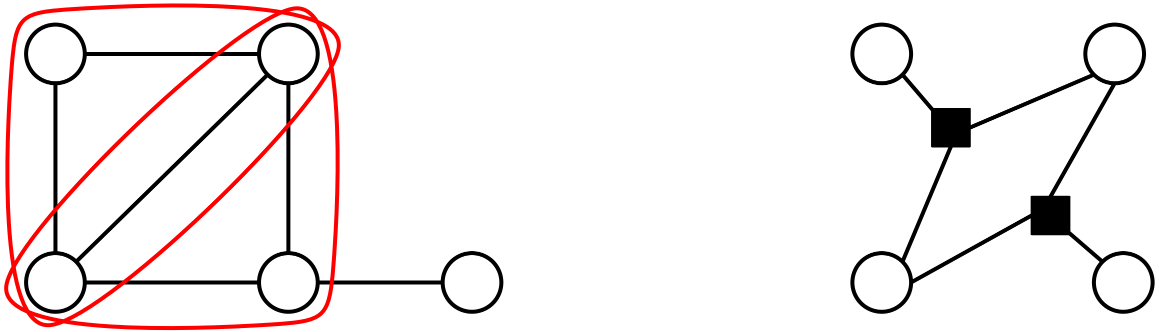
- Converting Markov Random Fields to factor graph takes the following steps:
  - Consider all the maximum cliques of the MRF
  - Create a factor node for each of the maximum cliques
  - Connect all the nodes of the maximum clique to the new factor nodes





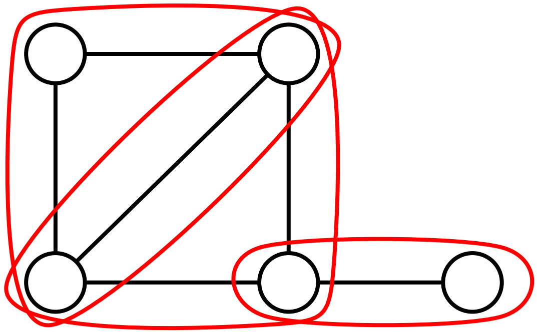
MRF ⟺ factor graph

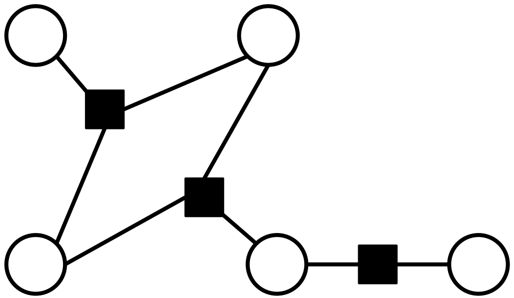
- Converting Markov Random Fields to factor graph takes the following steps:
  - Consider all the maximum cliques of the MRF
  - Create a factor node for each of the maximum cliques
  - Connect all the nodes of the maximum clique to the new factor nodes



MRF ⟺ factor graph

- Converting Markov Random Fields to factor graph takes the following steps:
  - Consider all the maximum cliques of the MRF
  - Create a factor node for each of the maximum cliques
  - Connect all the nodes of the maximum clique to the new factor nodes





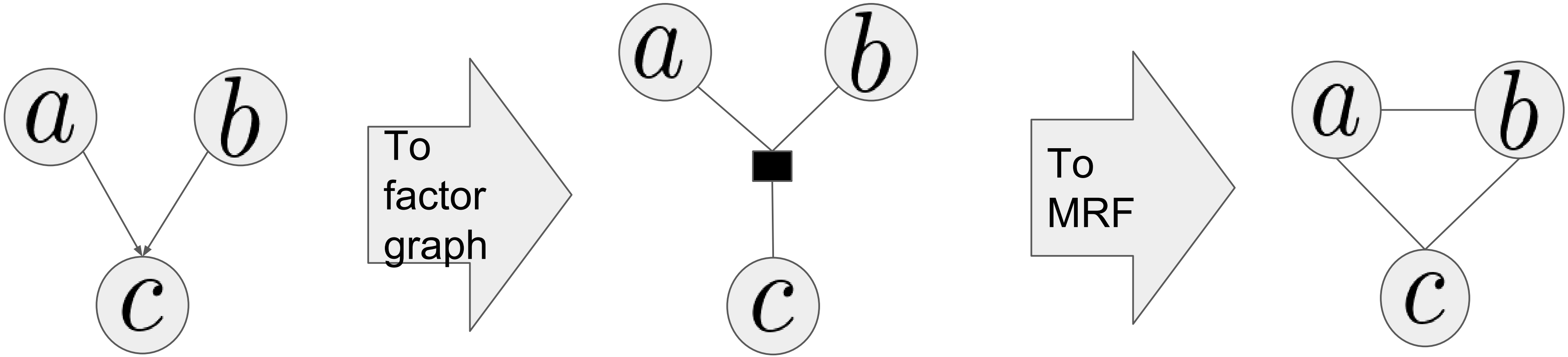
MRF ⟺ factor graph

- Convert FG back to MRF is easy
- For each factor, create all pairwise connections of the variables in the factor

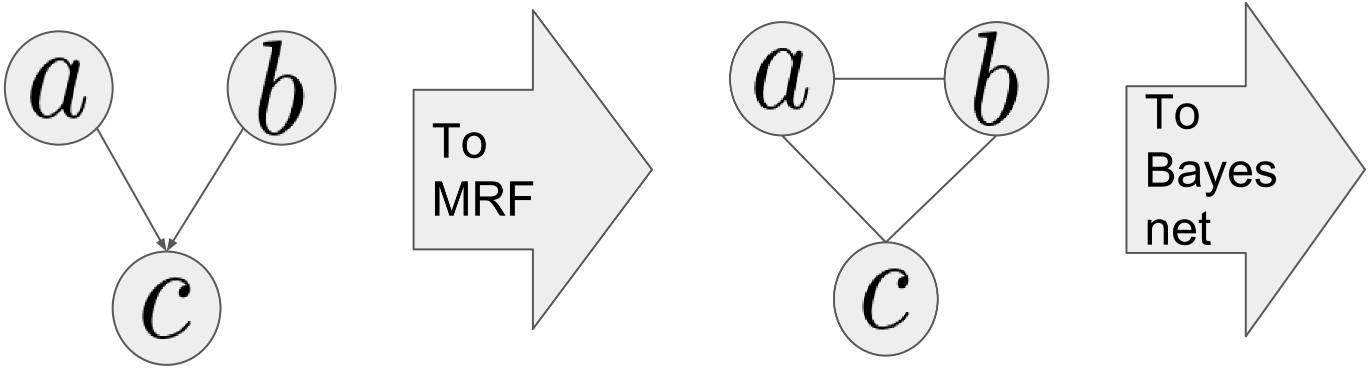
BNs $\Longleftrightarrow$ MRF

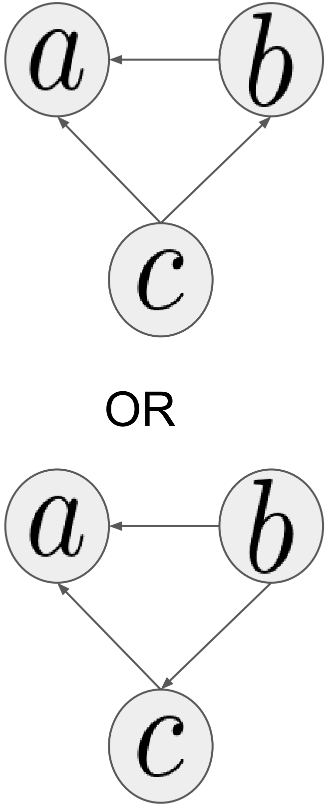
<u>Algorithm</u>:

- Create the factor graph for the Bayesian network
- Then remove the factors but add edges between any two nodes that share a factor



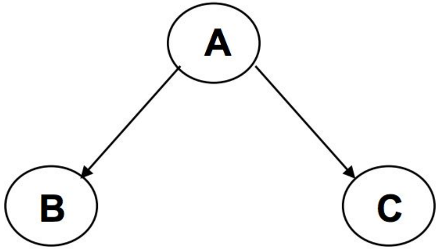
BNs $\Longleftrightarrow$ MRF





We don’t get the same Bayesian net back from this conversion...

Posterior inference example



Conditionally independent effects:
p(A,B,C) = p(B|A)p(C|A)p(A)

B and C are conditionally independent Given A

E.g., A is a disease, and we model B and C as conditionally independent symptoms given A

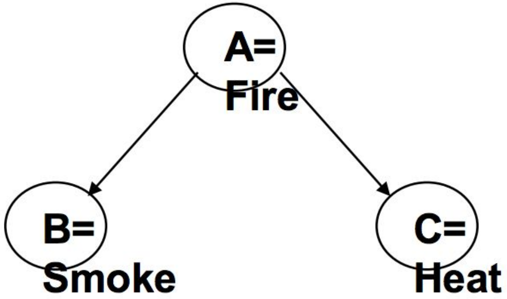
E.g., A is Fire, B is Heat, C is Smoke. "Where there's Smoke, there's Fire."

If we see Smoke, we can infer Fire.

If we see Smoke, observing Heat tells us very little additional information.

Posterior inference example

P(fire)
0.1



Fire	P(Smoke)
t	.90
f	.001

Fire	P(Heat)
t	.99
f	.0001

Conditionally independent effects:
P(A,B,C) = P(B|A)P(C|A)P(A)

Smoke and Heat are conditionally independent given Fire.

If we see B=Smoke, observing C=Heat tells us very little additional information.

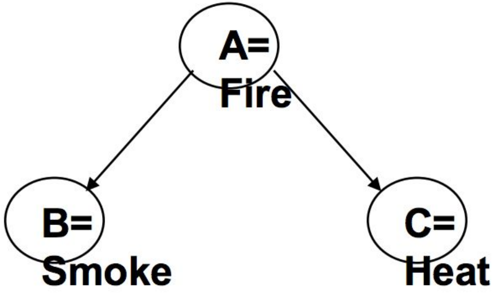
Posterior inference example

What is P(Fire=t | Smoke=t)?

P(Fire=t | Smoke=t)

=P(Fire=t & Smoke=t) / P(Smoke=t)

P(Fire)
0. 1

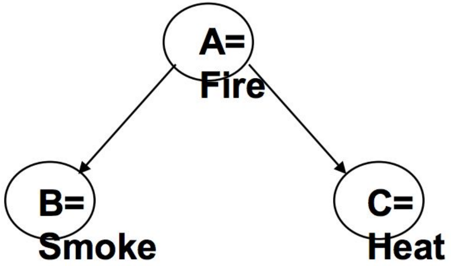


Fire	P(Smoke)
t	.90
f	.001

Fire	P(Heat)
t	.99
f	.0001

Posterior inference example

P(Fire)
0. 1



Fire	P(Smoke)
t	.90
f	.001

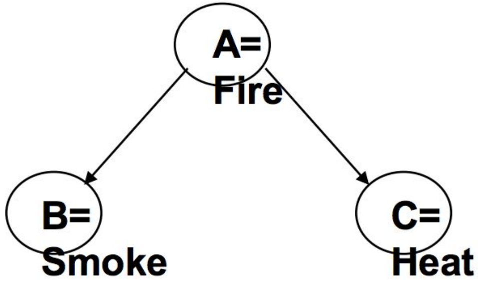
Fire	P(Heat)
t	.99
f	.0001

What is P(Fire=t & Smoke=t)?

P(Fire=t & Smoke=t)
=Σ_heat P(Fire=t&Smoke=t&heat)
=Σ_heat P(Smoke=t&heat|Fire=t)P(Fire=t)
=Σ_heat P(Smoke=t|Fire=t) P(heat|Fire=t)P(Fire=t)
=P(Smoke=t|Fire=t) P(heat=t|Fire=t)P(Fire=t)
+P(Smoke=t|Fire=t)P(heat=f|Fire=t)P(Fire=t)
= (.90x.99x.1)+(.90x.01x.1)
= 0.09

Posterior inference example

P(Fire)
0. 1



Fire	P(Smoke)
t	.90
f	.001

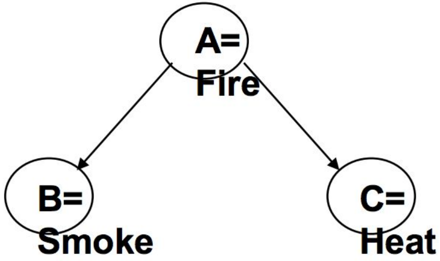
Fire	P(Heat)
t	.99
f	.0001

What is P(Smoke=t)?

$$
\begin{aligned}
&P(Smoke{=}t)\\
&=\Sigma\_fire\ \Sigma\_heat\ P(Smoke{=}t\ \&\ fire\ \&\ heat)\\
&=\Sigma\_fire\ \Sigma\_heat\ P(Smoke{=}t\ \&\ heat|fire)P(fire)\\
&=\Sigma\_fire\ \Sigma\_heat\ P(Smoke{=}t|fire)\ P(heat|fire)P(fire)\\
&=P(Smoke{=}t|fire{=}t)\ P(heat{=}t|fire{=}t)P(fire{=}t)\\
&\ +P(Smoke{=}t|fire{=}t)P(heat{=}f|fire{=}t)P(fire{=}t)\\
&\ +P(Smoke{=}t|fire{=}f)\ P(heat{=}t|fire{=}f)P(fire{=}f)\\
&\ +P(Smoke{=}t|fire{=}f)P(heat{=}f|fire{=}f)P(fire{=}f)\\
&= (.90x.99x.1)+(.90x.01x.1)\\
&\ +(.001x.0001x.9)+(.001x.9999x.9)\\
&\approx 0.0909
\end{aligned}
$$

Posterior inference example

P(Fire)
0. 1



Fire	P(Smoke)
t	.90
f	.001

Fire	P(Heat)
t	.99
f	.0001

What is P(Fire=t | Smoke=t)?

P(Fire=t | Smoke=t)
=P(Fire=t & Smoke=t) / P(Smoke=t)
≈ 0.09 / 0.0909
≈ 0.99

<u>So we've just proven that</u>
"Where there's smoke, there's (probably) fire."