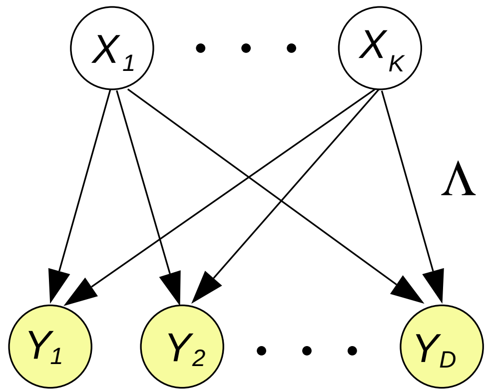
Statistical Approaches to Learning and Discovery

Graphical Models

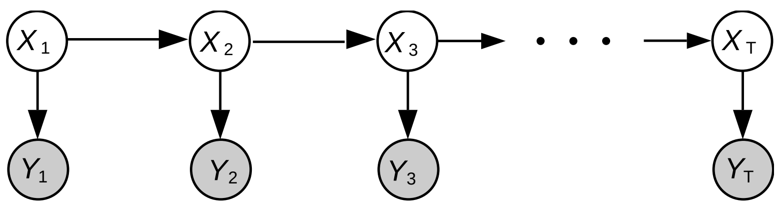
Zoubin Ghahramani & Teddy Seidenfeld
zoubin@cs.cmu.edu & teddy@stat.cmu.edu

CALD / CS / Statistics / Philosophy
Carnegie Mellon University
Spring 2002

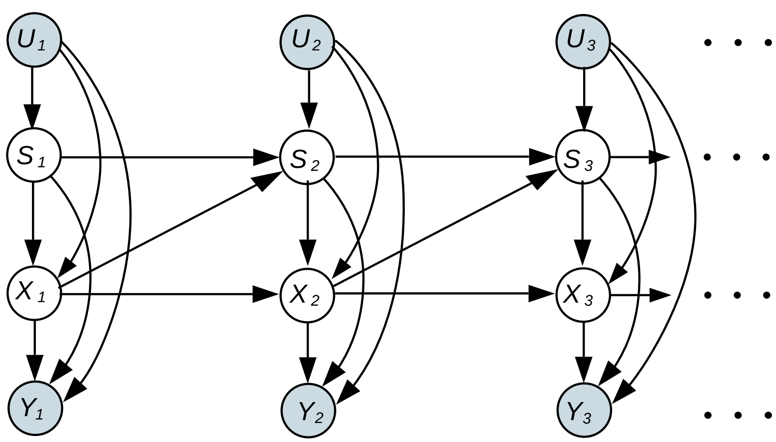
Some Examples



factor analysis
probabilistic PCA
ICA

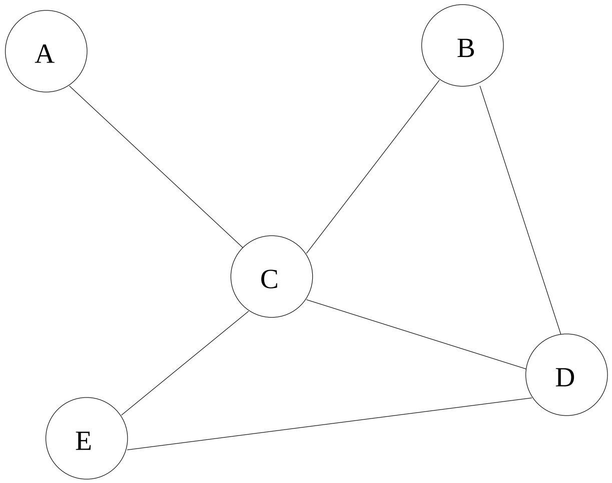


hidden Markov models
linear dynamical systems



switching state-space models

Markov Networks (Undirected Graphical Models)



Examples:
Boltzmann Machines
Markov Random Fields

Semantics: Every node is conditionally independent from its non-neighbors given its neighbors.

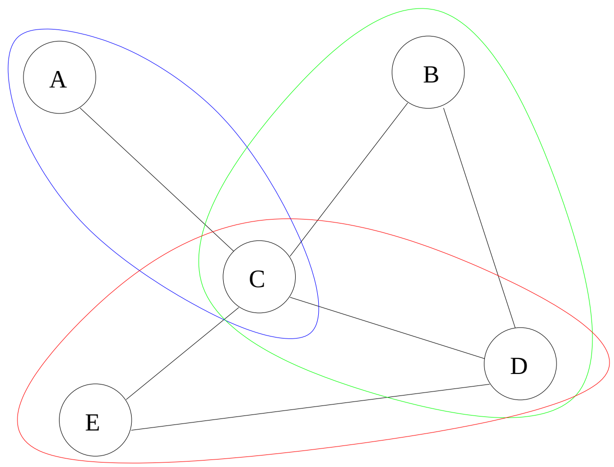
Conditional Independence: $X \perp\!\!\!\perp Y | V \Leftrightarrow p(X|Y,V) = p(X|V)$ when $p(Y,V) > 0$.
also $X \perp\!\!\!\perp Y | V \Leftrightarrow p(X,Y|V) = p(X|V)p(Y|V)$.

Markov Blanket: V is a Markov Blanket for X iff $X \perp\!\!\!\perp Y | V$ for all $Y \notin V$.

Markov Boundary: minimal Markov Blanket

Clique Potentials and Markov Networks

Definition: a *clique* is a fully connected subgraph (usually maximal).
C_i will denote the set of variables in the i^{th} clique.



1. Identify cliques of graph G

2. For each clique C_i assign a non-negative function $g_i(C_i)$ which measures "compatibility".

3. $p(X_1, \ldots, X_n) = \frac{1}{Z} \prod_i g_i(C_i)$ where $Z = \sum_{X_1 \cdots X_n} \prod_i g_i(C_i)$ is the normalization

The graph G embodies the conditional independencies in p (i.e. G is a Markov Field relative to p):
If V lies in *all* paths between X and Y in G, then $X \perp\!\!\!\perp Y | V$.

Hammersley–Clifford Theorem (1971)

Theorem: A probability function p formed by a normalized product of positive functions on cliques of G is a Markov Field relative to G.

Definition: The graph G is a *Markov Field relative to* p if it does not imply any conditional independence relationships that are not true in p.
(We are usually interested in the minimal such graph.)

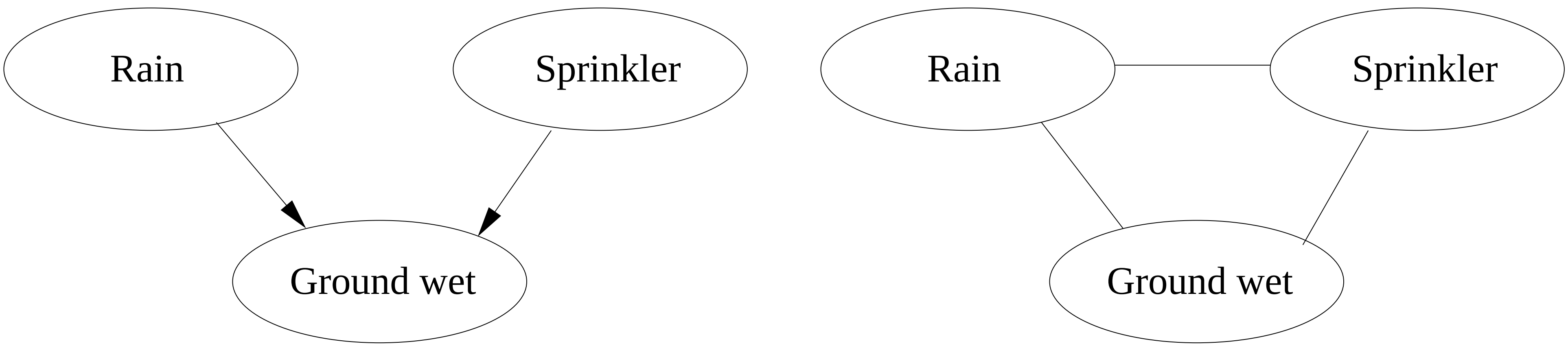
Proof: We need to show that the neighbors of X, $\mathrm{ne}(X)$ are a Markov Blanket for X:

$$
\begin{aligned}
p(X, Y, \ldots) =& \frac{1}{Z} \prod_i g_i(C_i) = \frac{1}{Z} \prod_{i:X \in C_i} g_i(C_i) \prod_{j:X \notin C_j} g_j(C_j) \\
=& \frac{1}{Z} f_1\big(X, \mathrm{ne}(X)\big)\ f_2\big(\mathrm{ne}(X), Y\big) = \frac{1}{Z'} p(X|\,\mathrm{ne}(X))\ p(Y|\,\mathrm{ne}(X))
\end{aligned}
$$

This shows that: $\quad p(X, Y|\,\mathrm{ne}(X)) = p(X|\,\mathrm{ne}(X))\ p(Y|\,\mathrm{ne}(X)) \Leftrightarrow X \perp\!\!\!\perp Y|\,\mathrm{ne}(X)$.

Problems with Markov Networks

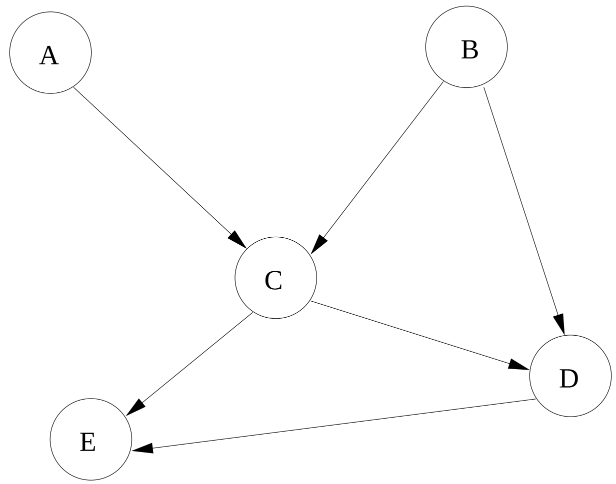
Many useful independencies are unrepresented — two variables are connected merely because some other variable depends on them:



Marginal independence vs. conditional independence.

“Explaining Away”

Bayesian Networks (Directed Graphical Models)



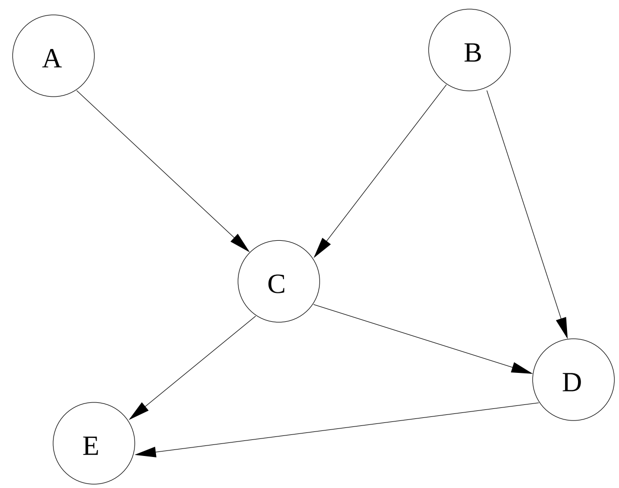
Semantics: $X \perp\!\!\!\perp Y | V$ if V d-separates X from Y.
Definition: V d-separates X from Y if along every undirected path between X and Y there is a node W such that either:

1. W has converging arrows along the path ($\rightarrow W \leftarrow$) and neither W nor its descendants are in V, or
2. W does not have converging arrows along the path ($\rightarrow W \rightarrow$) and $W \in V$.

The "Bayes-ball" algorithm.

Corollary: Markov Blanket for X: $\{\text{parents}(X) \cup \text{children}(X) \cup \text{parents-of-children}(X)\}$.

Bayesian Networks (Directed Graphical Models)



A Bayesian network corresponds to a factorization of the joint probability distribution:

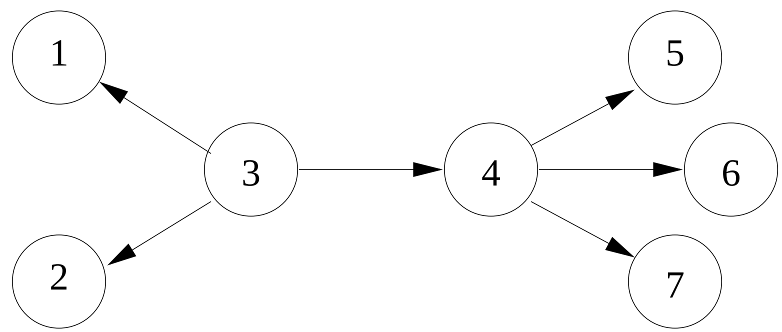
$$p(A, B, C, D, E) = p(A)p(B)p(C|A, B)p(D|B, C)p(E|C, D)$$

In general:

$$p(X_1, \dots, X_n) = \prod_{i=1}^{n} p(X_i | X_{\mathsf{pa}(i)})$$

where $\mathsf{pa}(i)$ are the parents of node i.

From Bayesian Trees to Markov Trees



$$p(1,2,\ldots,7) \;=\; p(3)p(1|3)p(2|3)p(4|3)p(5|4)p(6|4)p(7|4)$$

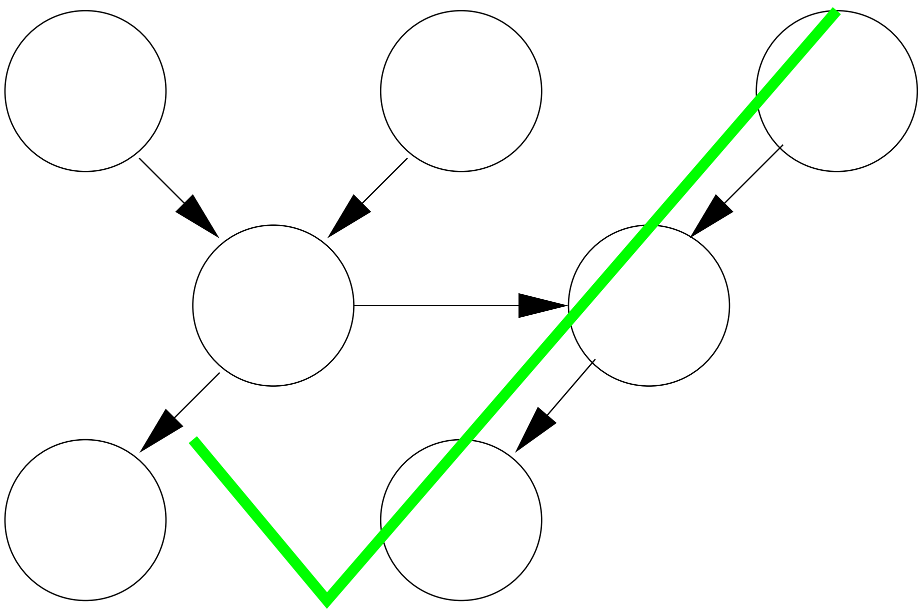
$$= \frac{p(1,3)p(2,3)p(3,4)p(4,5)p(4,6)p(4,7)}{p(3)p(3)p(4)p(4)p(4)}$$

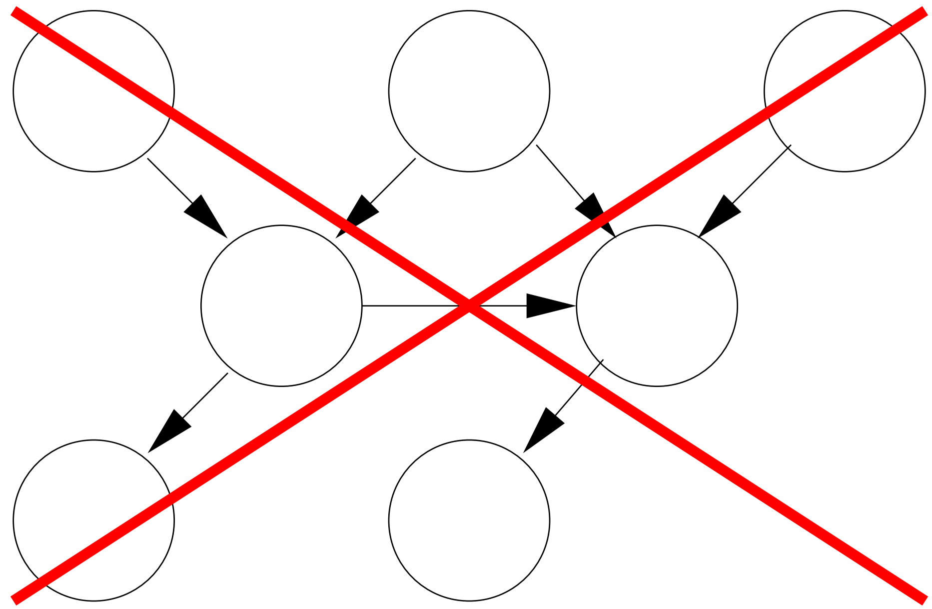
$$= \frac{\text{product of cliques}}{\text{product of clique intersections}}$$

$$= g(1,3)g(2,3)g(3,4)g(4,5)g(4,6)g(4,7) \;=\; \prod_i g_i(C_i)$$

Belief Propagation (in Singly Connected Bayesian Networks)

Definition: S.C.B.N. has an undirected underlying graph which is a tree, *ie* there is only one path between any two nodes.





Goal: For some node X we want to compute $p(X|e)$ given evidence e.
Since we are considering S.C.B.N.s:

- every node X divides the evidence into upstream e_X^+ and downstream e_X^-
- every arc $X \rightarrow Y$ divides the evidence into upstream e_{XY}^+ and downstream e_{XY}^-.

The three key ideas behind Belief Propagation

Idea 1: Our belief about the variable X can be found by combining upstream and downstream evidence:

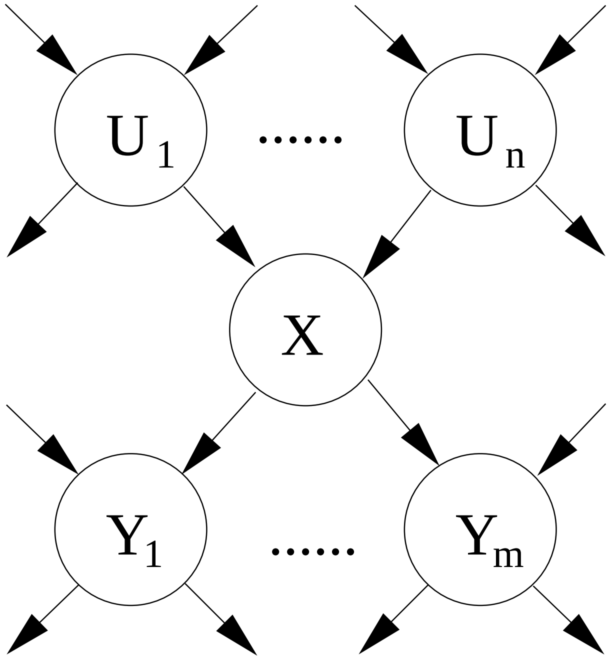
$$p(X|e) = \frac{p(X,e)}{p(e)} = \frac{p(X,e_X^+,e_X^-)}{p(e_X^+,e_X^-)} \propto p(X|e_X^+) \quad \times \quad \underbrace{p(e_X^-|X,e_X^+)}_{X \text{ d-separates } e_X^- \text{ from } e_X^+}$$

$$= p(X|e_X^+)p(e_X^-|X) = \pi(X)\lambda(X)$$

Idea 2: The upstream and downstream evidence can be computed via a local message passing algorithm between the nodes in the graph.

Idea 3: "Don't send back to a node (any part of) the message it sent to you!"

Belief Propagation



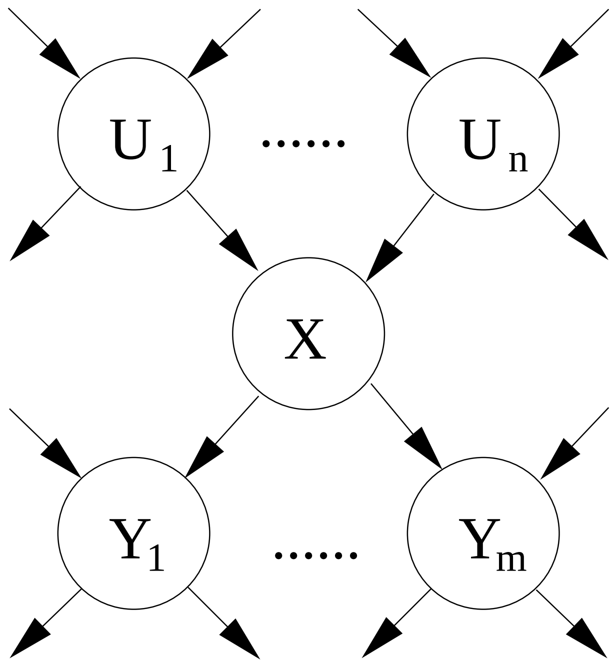
top-down causal support:
$\pi_X(U_i) = p(U_i | e^+_{U_i X})$

bottom-up diagnostic support:
$\lambda_{Y_j}(X) = p(e^-_{X Y_j} | X)$

To update the belief about X:

$$\begin{aligned}
\mathrm{BEL}(X) &= \frac{1}{Z}\lambda(X)\pi(X) \\
\lambda(X) &= \prod_j \lambda_{Y_j}(X) \\
\pi(X) &= \sum_{U_1 \cdots U_n} p(X | U_1, \ldots, U_n) \prod_i \pi_X(U_i)
\end{aligned}$$

Belief Propagation, cont



top-down causal support:
$\pi_X(U_i) = p(U_i | e^+_{U_i X})$

bottom-up diagnostic support:
$\lambda_{Y_j}(X) = p(e^-_{X Y_j} | X)$

Bottom-up propagation, X sends to U_i:

$$\lambda_X(U_i) \; = \; \frac{1}{Z} \sum_X \lambda(X) \sum_{U_k : k \neq i} p(X | U_1, \ldots, U_n) \prod_{k \neq i} \pi_X(U_k)$$

Top-down propagation, X sends to Y_j:

$$\pi_{Y_j}(X) \; = \; \frac{1}{Z} [\prod_{k \neq j} \lambda_{Y_k}(X)] \sum_{U_1 \cdots U_n} p(X | U_1, \ldots, U_n) \prod_i \pi_X(U_i) \; = \; \frac{1}{Z} \frac{\mathrm{BEL}(X)}{\lambda_{Y_j}(X)}$$

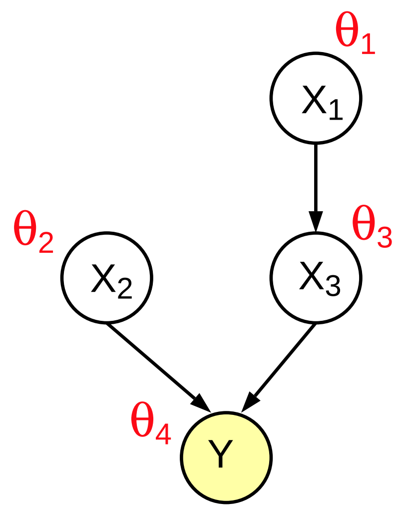
Belief Propagation in multiply connected Bayesian Networks

The Junction Tree algorithm: Form an undirected graph from your directed graph such that no additional conditional independence relationships have been created (this step is called "moralization"). Lump variables in cliques together and form a tree of cliques—this may require a nasty step called "triangulation". Do inference in this tree.

Cutset Conditioning: or "reasoning by assumptions". Find a small set of variables which, if they were given (i.e. known) would render the remaining graph singly connected. For each value of these variables run belief propagation on the singly connected network. Average the resulting beliefs with the appropriate weights.

Loopy Belief Propagation: just use BP although there are loops. In this case the terms "upstream" and "downstream" are not clearly defined. No guarantee of convergence, but often works well in practice.

Learning with Hidden Variables: The EM Algorithm



Assume a model parameterised by θ with observable variables Y and hidden variables X

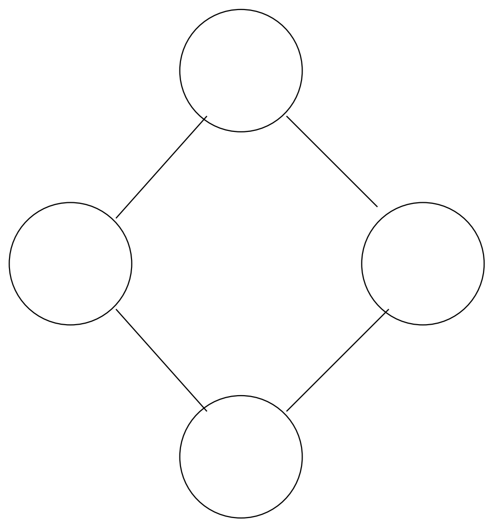
Goal: maximise log likelihood of observables.

$$\mathcal{L}(\theta) = \ln p(Y|\theta) = \ln \sum_X p(Y, X|\theta)$$

- **E-step**: first infer $p(X|Y, \theta_{old})$, then
- **M-step**: find θ_{new} using complete data learning

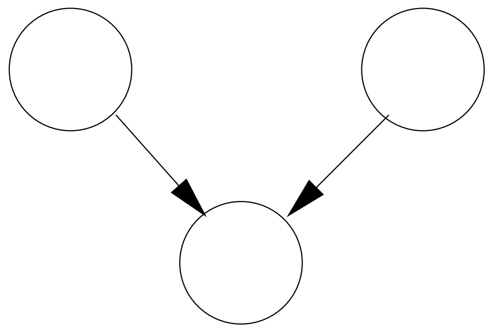
The E-step requires solving the *inference* problem: finding explanations, X, for the data, Y, given the current model, θ (using e.g. BP).

Expressive Power of Bayesian and Markov Networks



No Bayesian network can represent these and only these independencies

No matter how we direct the arrows there will always be two non-adjacent parents sharing a common child $\Longrightarrow$ dependence in Bayesian network but independence in Markov network.



No Markov network can represent these and only these independencies