4F13: Machine Learning

Lectures 6-7: Graphical Models

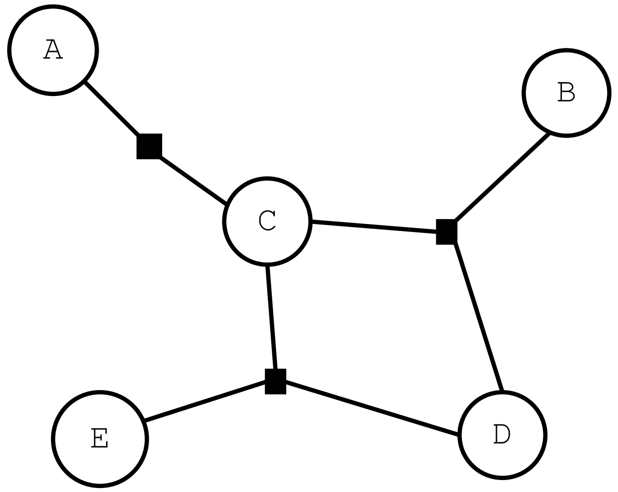
Zoubin Ghahramani
`zoubin@eng.cam.ac.uk`

Department of Engineering
University of Cambridge

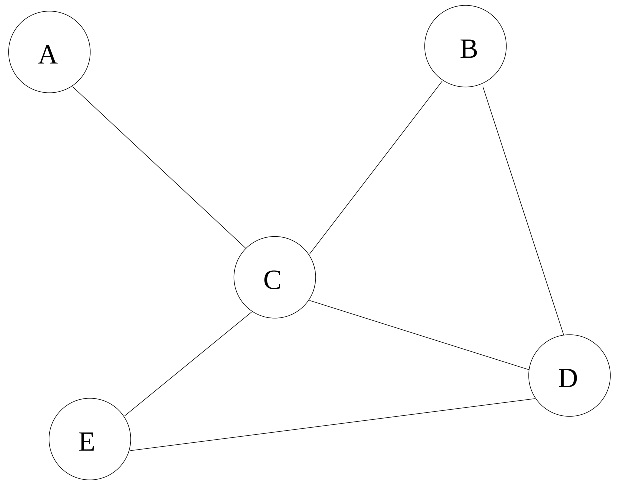
Michaelmas, 2006

http://learning.eng.cam.ac.uk/zoubin/ml06/

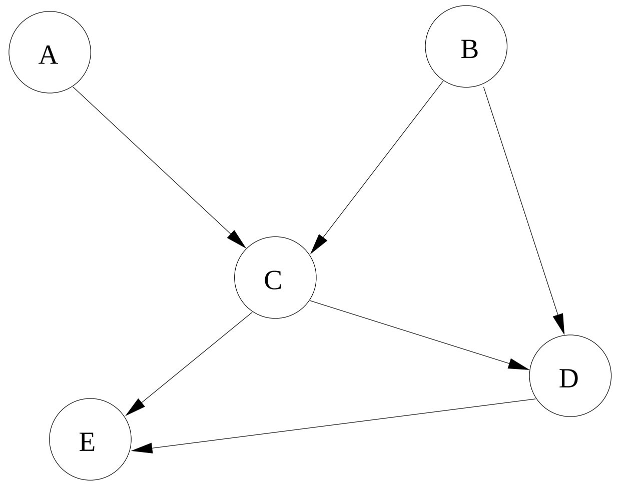
Three main kinds of graphical models



factor graph



undirected graph



directed graph

- Nodes correspond to random variables
- Edges represent statistical dependencies between the variables

Why do we need graphical models?

- Graphs are an **intuitive** way of representing and visualising the relationships between many variables. (Examples: family trees, electric circuit diagrams, neural networks)

- A graph allows us to abstract out the **conditional independence** relationships between the variables from the details of their parametric forms. Thus we can ask questions like: "Is A dependent on B given that we know the value of C ?" just by looking at the graph.

- Graphical models allow us to define general **message-passing algorithms** that implement Bayesian inference efficiently. Thus we can answer queries like "What is $P(A|C = c)$?" without enumerating all settings of all variables in the model.

Graphical models $=$ statistics $\times$ graph theory $\times$ computer science.

Conditional Independence

Conditional Independence:

$$X \perp\!\!\!\perp Y | V \;\;\Leftrightarrow\;\; p(X|Y,V) = p(X|V)$$

when $p(Y,V) > 0$. Also

$$X \perp\!\!\!\perp Y | V \;\;\Leftrightarrow\;\; p(X,Y|V) = p(X|V)\,p(Y|V)$$

In general we can think of conditional independence between **sets of variables**:

$$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{V} \;\;\Leftrightarrow\;\; \{X \perp\!\!\!\perp Y | \mathcal{V},\;\; \forall X \in \mathcal{X} \;\text{ and }\; \forall Y \in \mathcal{Y}\}$$

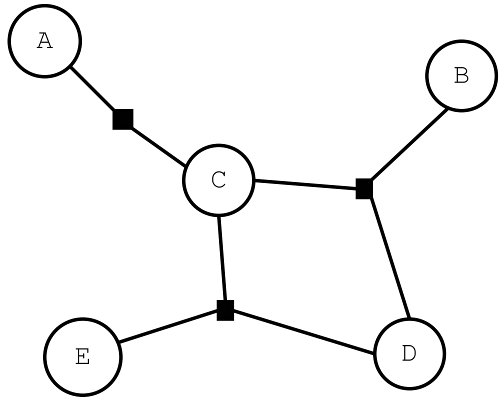
Marginal Independence:

$$X \perp\!\!\!\perp Y \;\;\Leftrightarrow\;\; X \perp\!\!\!\perp Y | \emptyset \;\;\Leftrightarrow\;\; p(X,Y) = p(X)\,p(Y)$$

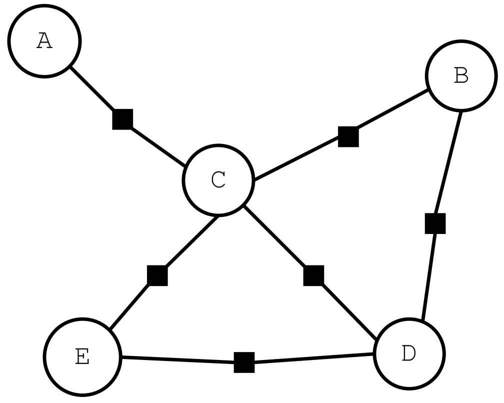
Conditional and Marginal Independence (Examples)

- Amount of Speeding Fine $\perp\!\!\!\perp$ Type of Car | Speed
- Lung Cancer $\perp\!\!\!\perp$ Yellow Teeth | Smoking
- (Position, Velocity)$_{t+1}$ $\perp\!\!\!\perp$ (Position, Velocity)$_{t-1}$ | (Position, Velocity)$_t$, Acceleration$_t$
- Child's Genes $\perp\!\!\!\perp$ Grandparents' Genes | Parents' Genes (approximately)
- Ability of Team A $\perp\!\!\!\perp$ Ability of Team B
- not (Ability of Team A $\perp\!\!\!\perp$ Ability of Team B | Outcome of A vs B Game)

Factor Graphs



(a)



(b)

Two types of nodes:

- The circles in a factor graph represent random variables (e.g. A).
- The filled dots represent factors in the joint distribution (e.g. $g_1(\cdot)$).

(a) $P(A,B,C,D,E) = \frac{1}{Z} g_1(A,C) g_2(B,C,D) g_3(C,D,E)$

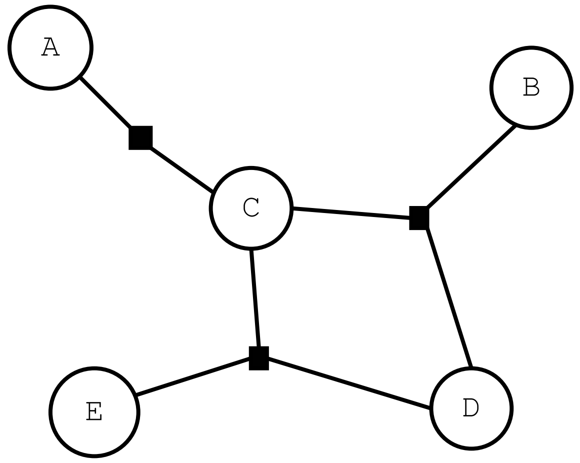
(b) $P(A,B,C,D,E) = \frac{1}{Z} g_1(A,C) g_2(B,C) g_3(C,D) g_4(B,D) g_5(C,E) g_6(D,E)$

The g_i are non-negative functions of their arguments, and Z is a normalization constant. E.g. in (a), if all variables are discrete and take values in $\mathcal{A} \times \mathcal{B} \times \mathcal{C} \times \mathcal{D} \times \mathcal{E}$:

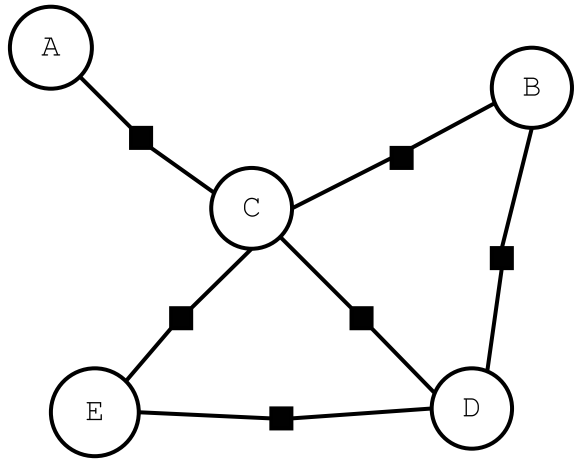
$$Z = \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} \sum_{c \in \mathcal{C}} \sum_{d \in \mathcal{D}} \sum_{e \in \mathcal{E}} g_1(A=a, C=c) g_2(B=b, C=c, D=d) g_3(C=c, D=d, E=e)$$

Two nodes are neighbors if they share a common factor.

Factor Graphs



(a)



(b)

The circles in a factor graph represent random variables.
The filled dots represent factors in the joint distribution.

(a) $P(A, B, C, D, E) = \frac{1}{Z} g_1(A, C) g_2(B, C, D) g_3(C, D, E)$

(b) $P(A, B, C, D, E) = \frac{1}{Z} g_1(A, C) g_2(B, C) g_3(C, D) g_4(B, D) g_5(C, E) g_6(D, E)$

Two nodes are neighbors if they share a common factor.

Definition: A *path* is a sequence of neighboring nodes.

Fact: $X \perp\!\!\!\perp Y | \mathcal{V}$ if every path between X and Y contains some node $V \in \mathcal{V}$

Corollary: Given the neighbors of X, the variable X is conditionally independent of all other variables: $X \perp\!\!\!\perp Y | \operatorname{ne}(X), \quad \forall Y \notin \{X \cup \operatorname{ne}(X)\}$

Proving Conditional Independence

Conditional independence:

$$X \perp\!\!\!\perp Y | V \quad \Leftrightarrow \quad p(X|Y,V) = p(X|V) \tag{1}$$

Assume:

$$P(X,Y,V) = \frac{1}{Z} g_1(X,V) g_2(Y,V) \tag{2}$$

Then summing (2) over X we get:

$$P(Y,V) = \frac{1}{Z} [\sum_X g_1(X,V)] g_2(Y,V) \tag{3}$$

Dividing (2) by (3) we get:

$$P(X|Y,V) = \frac{g_1(X,V)}{\sum_X g_1(X,V)} \tag{4}$$

Since the rhs. of (4) doesn't depend on Y, it follows that X is independent of Y given V. Therefore factorizaton (2) implies conditional independence (1).

Undirected Graphical Models

In an Undirected Graphical Model, the joint probability over all variables can be written in a factored form:

$$P(\mathbf{x}) = \frac{1}{Z} \prod_j g_j(\mathbf{x}_{C_j})$$

where $\mathbf{x} = (x_1, \ldots, x_K)$, and

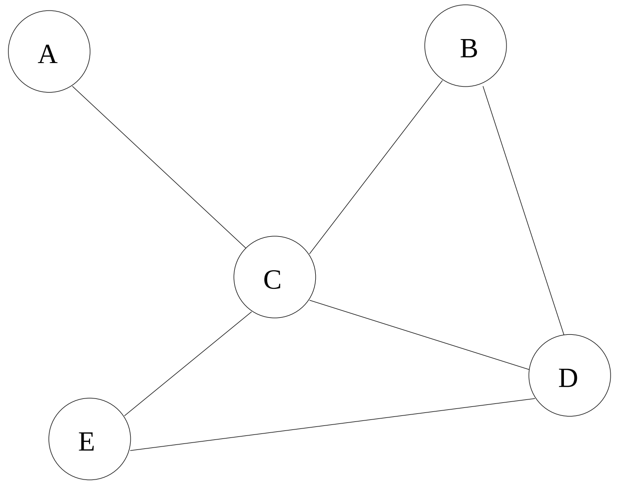
$$C_j \subseteq \{1, \ldots, K\}$$

are subsets of the set of all variables, and $\mathbf{x}_S \equiv (x_k : k \in S)$.

Graph Definition: Let each variable be a node. Connect nodes i and k if there exists a set C_j such that both $i \in C_j$ and $k \in C_j$. These sets form the *cliques* of the graph (fully connected subgraphs).

Note: Undirected Graphical Models are also called *Markov Networks*.
Very similar to factor graphs.

Undirected Graphical Models



$$P(A,B,C,D,E) = \frac{1}{Z} g_1(A,C) g_2(B,C,D) g_3(C,D,E)$$

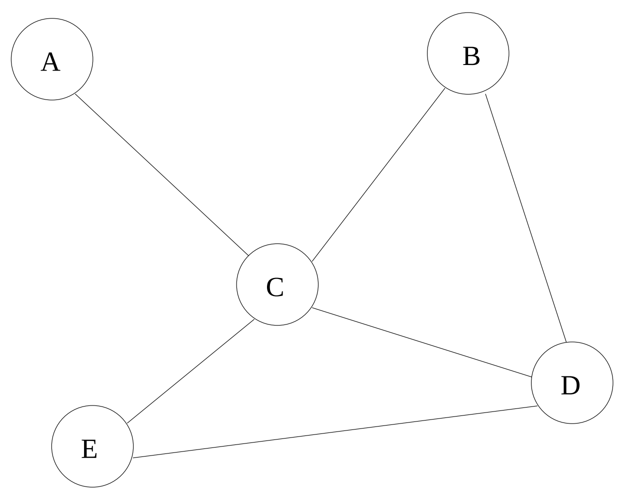
Fact: $X \perp\!\!\!\perp Y | \mathcal{V}$ if every path between X and Y contains some node $V \in \mathcal{V}$

Corollary: Given the neighbors of X, the variable X is conditionally independent of all other variables: $X \perp\!\!\!\perp Y \,|\, \mathrm{ne}(X), \quad \forall Y \notin \{X \cup \mathrm{ne}(X)\}$

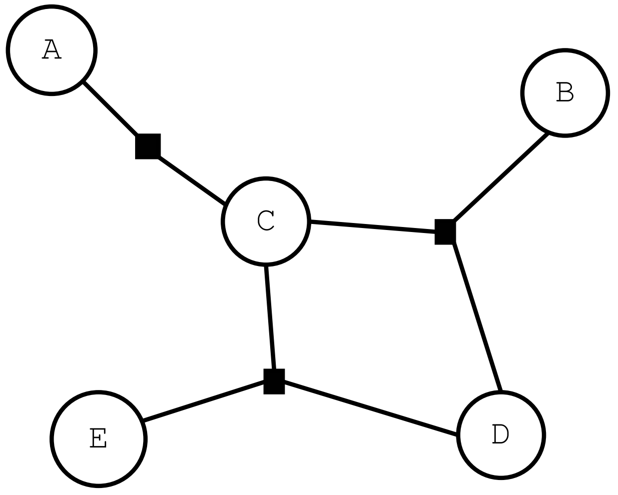
Markov Blanket: $\mathcal{V}$ is a Markov Blanket for X iff $X \perp\!\!\!\perp Y | \mathcal{V}$ for all $Y \notin \{X \cup \mathcal{V}\}$.

Markov Boundary: minimal Markov Blanket $\equiv \mathrm{ne}(X)$ for undirected graphs and factor graphs

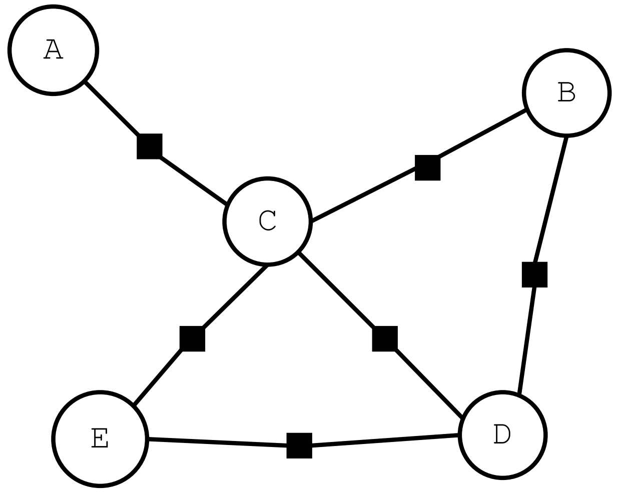
Comparing Undirected Graphs and Factor Graphs



(a)



(b)



(c)

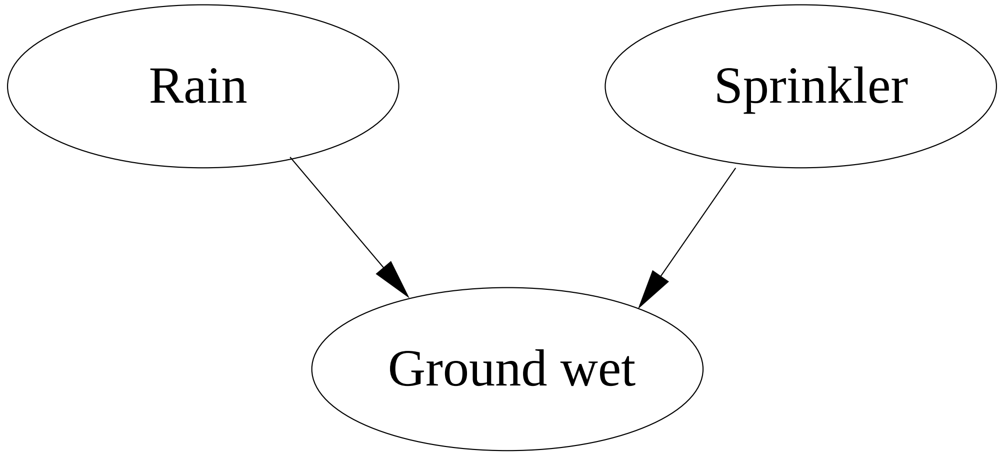
All nodes in (a), (b), and (c) have exactly the same neighbors and therefore these three graphs represent exactly the same conditional independence relationships.

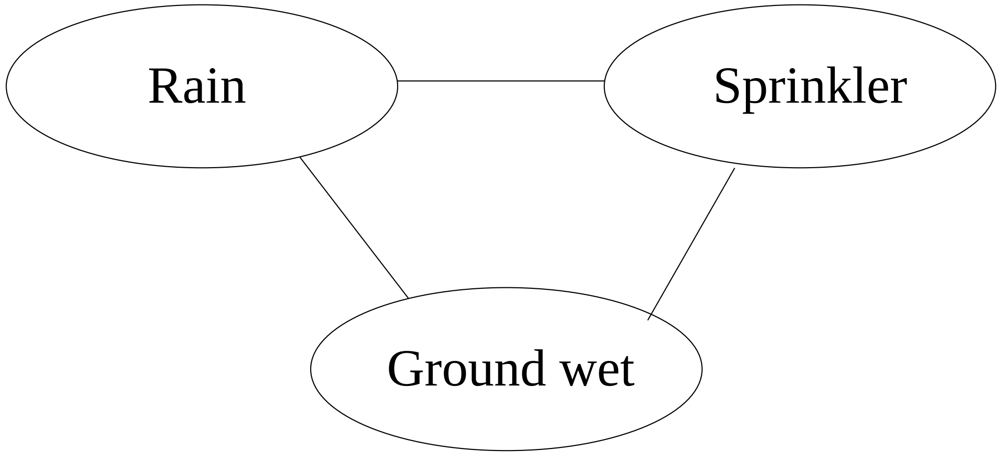
(c) also represents the fact that the probability factors into a product of pairwise functions.

Consider the case where each variables is discrete and can take on K possible values. Then the functions in (a) and (b) are tables with $\mathcal{O}(K^3)$ cells, whereas in (c) they are $\mathcal{O}(K^2)$.

Problems with Undirected Graphs and Factor Graphs

In UGs and FGs, many useful independencies are unrepresented—two variables are connected merely because some other variable depends on them:



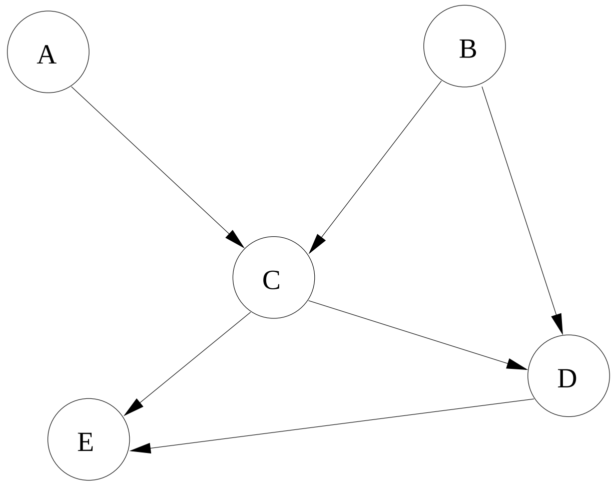


This highlights the difference between **marginal independence** and **conditional independence**.

R and S are marginally independent (i.e. given nothing), but they are conditionally dependent given G

"Explaining Away": Observing that the spinkler is on, explains away the fact that the ground was wet, therefore we don't need to believe that it rained.

Directed Acyclic Graphical Models (Bayesian Networks)



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

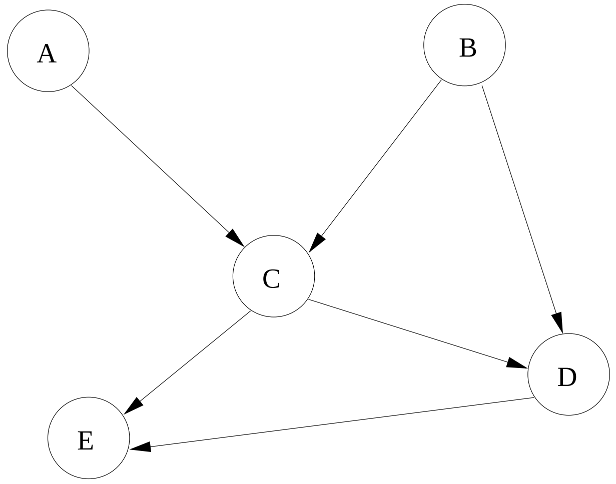
$$p(A, B, C, D, E) = p(A)p(B)p(C|A, B)p(D|B, C)p(E|C, D)$$

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.

Directed Acyclic Graphical Models (Bayesian Networks)



Semantics: $X \perp\!\!\!\perp Y | \mathcal{V}$ if $\mathcal{V}$ d-separates X from Y[1].

Definition: $\mathcal{V}$ d-separates X from Y if *every* undirected path[2] between X and Y is **blocked** by $\mathcal{V}$. A path is blocked by $\mathcal{V}$ if there is a node W on the path such that either:

1. W has converging arrows along the path ($\rightarrow W \leftarrow$)[3] and neither W nor its descendants are in $\mathcal{V}$, or
2. W does not have converging arrows along the path ($\rightarrow W \rightarrow$ or $\leftarrow W \rightarrow$) and $W \in \mathcal{V}$.

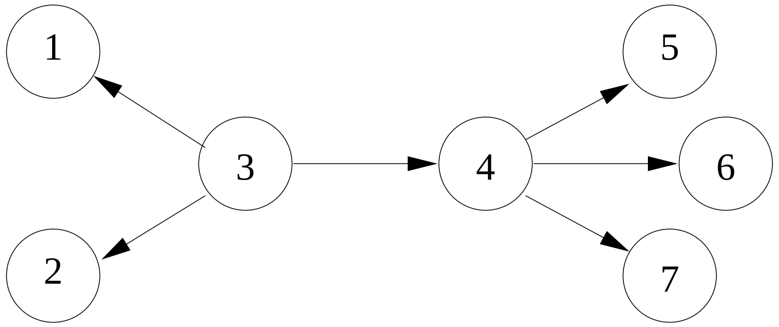
Corollary: Markov Boundary for X: $\{\text{parents}(X) \cup \text{children}(X) \cup \text{parents-of-children}(X)\}$.

[1] See also the "Bayes Ball" algorithm in the Appendix

[2] An undirected path ignores the direction of the edges.

[3] Note that converging arrows *along the path* only refers to what happens on that path. Also called a *collider*.

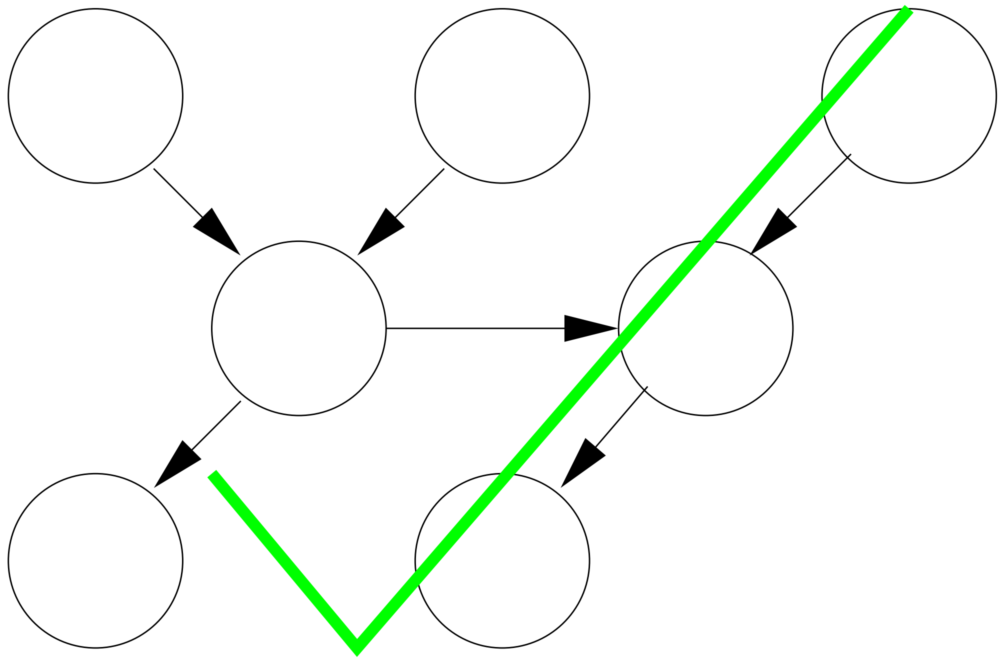
From Directed Trees to Undirected Trees

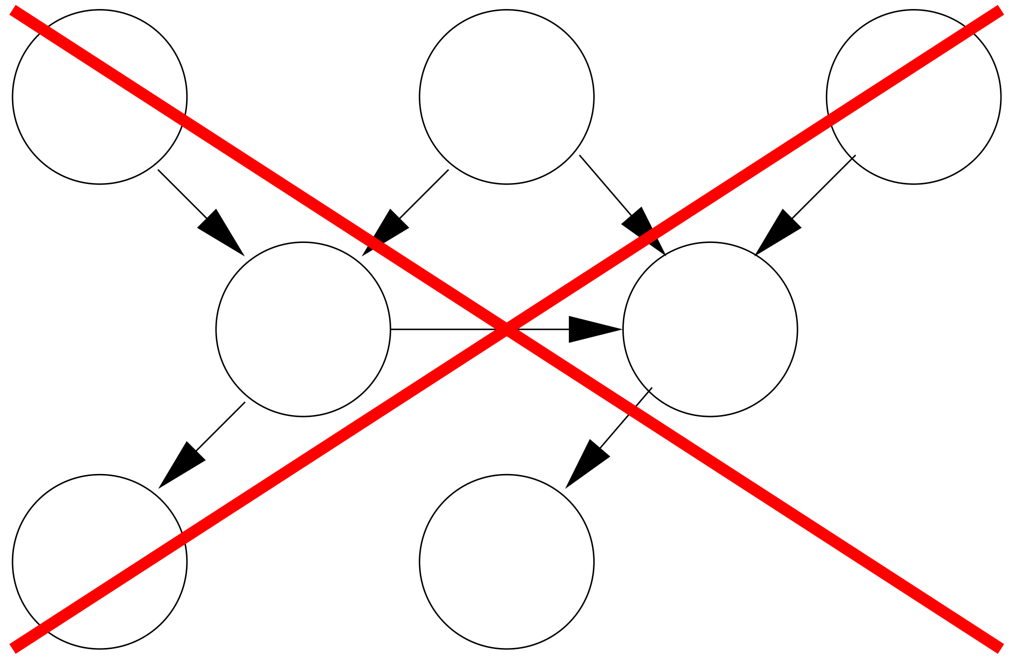


$$
\begin{aligned}
p(x_1, x_2, \ldots, x_7) &= p(x_3)p(x_1|x_3)p(x_2|x_3)p(x_4|x_3)p(x_5|x_4)p(x_6|x_4)p(x_7|x_4) \\
&= \frac{p(x_1, x_3)p(x_2, x_3)p(x_3, x_4)p(x_4, x_5)p(x_4, x_6)p(x_4, x_7)}{p(x_3)p(x_3)p(x_4)p(x_4)p(x_4)} \\
&= \frac{\text{product of cliques}}{\text{product of clique intersections}} \\
&= g_1(x_1, x_3)g_2(x_2, x_3)g_3(x_3, x_4)g_4(x_4, x_5)g_5(x_4, x_6)g_6(x_4, x_7) = \\
&= \prod_i g_i(C_i)
\end{aligned}
$$

Belief Propagation (in singly connected DAGs)

Definition: A DAG is *singly connected* if its underlying undirected graph is a tree, *ie* there is only one undirected path between any two nodes.

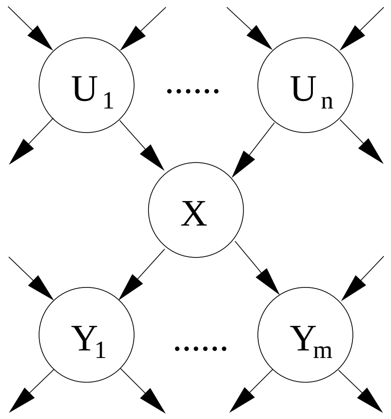




Goal: For some node X we want to compute $p(X|e)$ given evidence e.
Since we are considering singly connected graphs:

- every node X divides the evidence into upstream e^+_X and downstream e^-_X
- every edge $X \to Y$ 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:

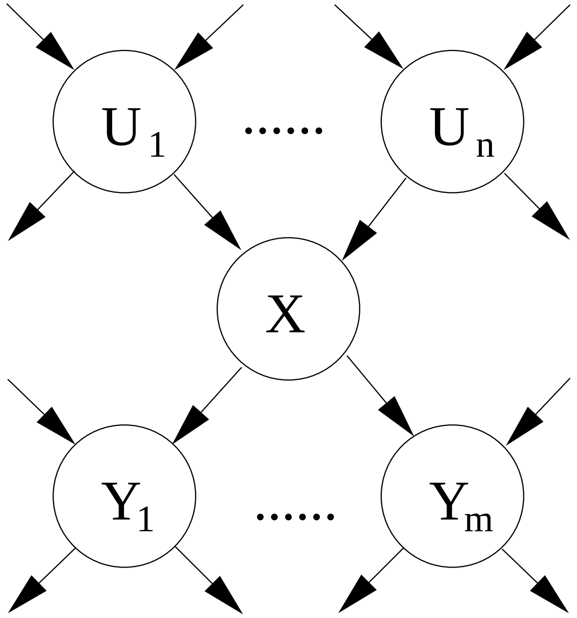
$$p(X|e) = \frac{p(X,e)}{p(e)} = \frac{p(X,e_X^+,e_X^-)}{p(e_X^+,e_X^-)} \propto p(X|e_X^+) \quad \times \quad \underbrace{p(e_X^-|X,e_X^+)}_{X \text{ d-separates } e_X^- \text{ from } e_X^+}$$

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

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



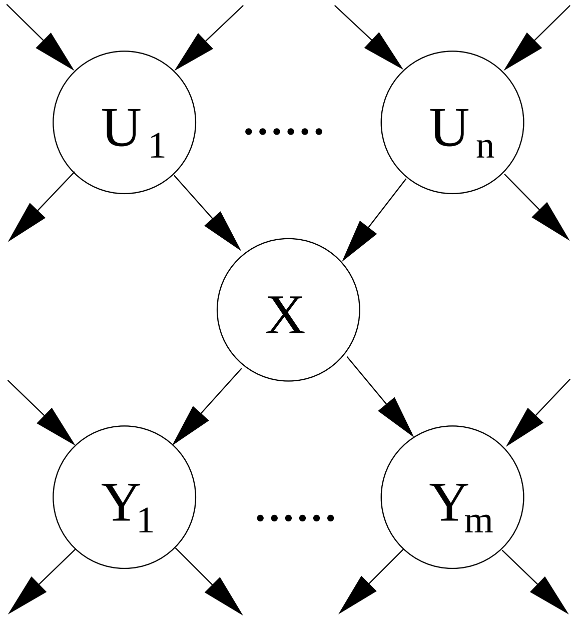
top-down upstream evidence:
$\pi_X(U_i) = p(U_i|e^+_{U_iX})$

bottom-up downstream evidence:
$\lambda_{Y_j}(X) = p(e^-_{XY_j}|X)$

To update the belief about X given the evidence:

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

Belief Propagation (cont.)



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

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

Bottom-up propagation, message X sends to U_i:

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

Top-down propagation, message X sends to Y_j:

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

Z is the normaliser ensuring $\sum_X \pi_{Y_j}(X) = 1$

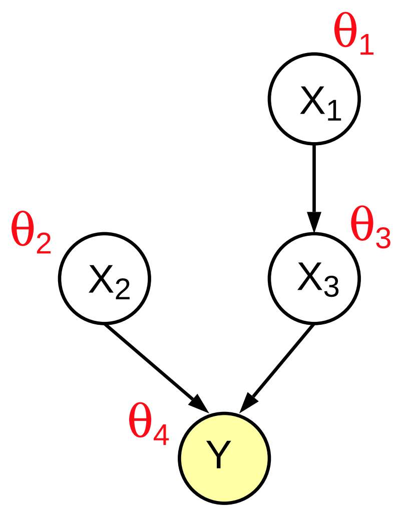
Belief Propagation in multiply connected Bayesian Networks

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 parameter log likelihood given observables.

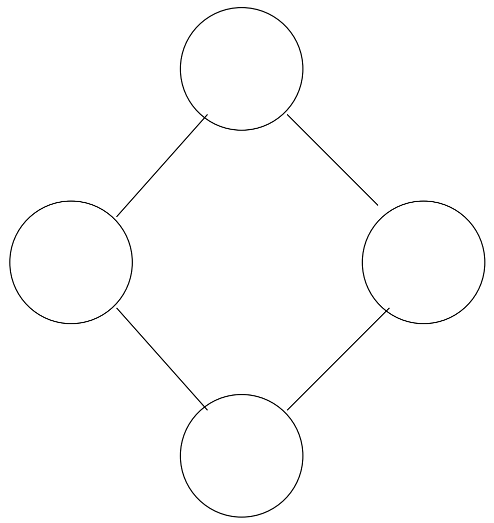
$$\mathcal{L}(\theta) = \ln p(Y|\theta) = \ln \sum_X p(Y, X|\theta)$$

- **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).

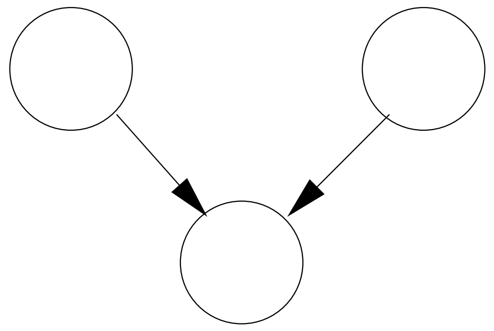
How about structure learning?

Expressive Power of Directed and Undirected Graphs



No Directed Graph (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 Directed Graph but independence in Undirected Graph.

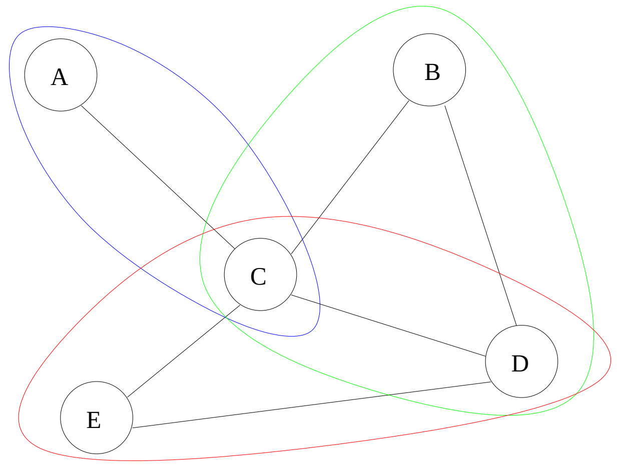


No Undirected Graph or Factor Graph can represent these and only these independencies

Appendix: Clique Potentials and Undirected Graphs

Definition: a *clique* is a fully connected subgraph. By clique we usually mean maximal clique (i.e. not contained within another clique)

C_i denotes the set of variables in the i^{th} clique.



$$p(x_1, \ldots, x_K) = \frac{1}{Z} \prod_i g_i(\mathbf{x}_{C_i})$$

where $Z = \sum_{x_1 \cdots x_K} \prod_i g_i(\mathbf{x}_{C_i})$ is the normalization.

Associated with each clique C_i is a non-negative function $g_i(\mathbf{x}_{C_i})$ which measures "compatibility" between settings of the variables.

A	C	$g_1(A,C)$
0	0	0.2
0	1	0.6
1	0	0.0
1	1	1.2

Example: Let $C_1 = \{A, C\}, A \in \{0,1\}, C \in \{0,1\}$
What does this mean?

Appendix: 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 distribution p is a *Markov Field relative to* G if all conditional independence relations represented by G are true of p.

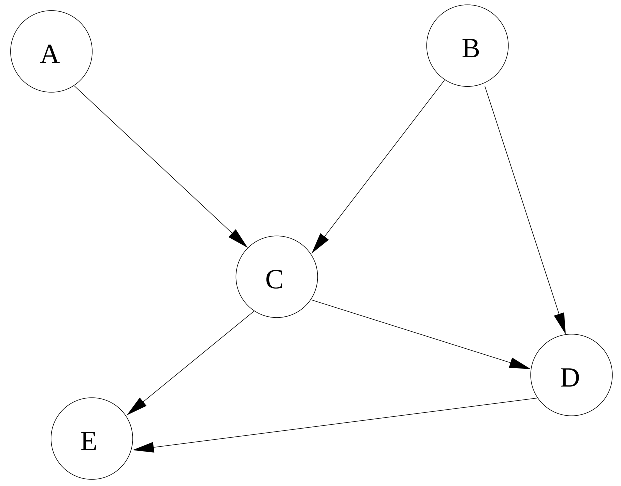
G represents the following CI relations: If $V \in \mathcal{V}$ lies on *all* paths between X and Y in G, then $X \perp\!\!\!\perp Y | \mathcal{V}$.

Proof: We need to show that if p is a product of functions on cliques of G then a variable is conditionally independent of its non-neighbors in G given its neighbors in G. That is: $\mathrm{ne}(x_\ell)$ is a Markov Blanket for x_ℓ. Let $x_m \notin \{x_\ell \cup \mathrm{ne}(x_\ell)\}$

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

It follows that: $p(x_\ell, x_m | \,\mathrm{ne}(x_\ell)) = p(x_\ell | \,\mathrm{ne}(x_\ell))\ p(x_m | \,\mathrm{ne}(x_\ell)) \Leftrightarrow x_\ell \perp\!\!\!\perp x_m | \,\mathrm{ne}(x_\ell)$.

Appendix: The "Bayes-ball" algorithm



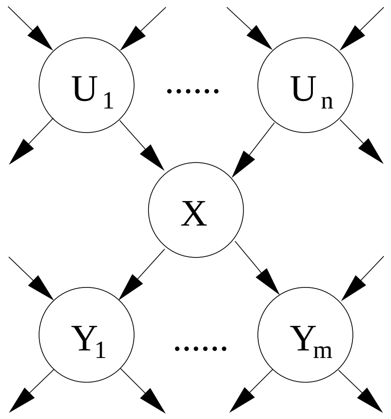
Game: can you get a ball from X to Y without being blocked by $\mathcal{V}$?

Depending on the direction the ball came from and the type of node, the ball can **pass through** (from a parent to all children, from a child to all parents), **bounce back** (from any parent to all parents, or from any child to all children), or be **blocked**.

- An unobserved (hidden) node ($W \notin \mathcal{V}$) passes balls through but also bounces back balls from children.

- An observed (given) node ($W \in \mathcal{V}$) bounces back balls from parents but blocks balls from children.

Appendix: Understanding BP equations



$$p(X|e) = \mathrm{BEL}(X) = \frac{1}{Z}\lambda(X)\pi(X) = p(e_X^-|X)p(X|e_X^+) \quad (5)$$

$$p(e_X^-|X) = \lambda(X) = \prod_j \lambda_{Y_j}(X) = \prod_j P(e_{XY_j}^-|X) \quad (6)$$

$$p(X|e_X^+) = \pi(X) = \sum_{U_1 \cdots U_n} p(X|U_1,\ldots,U_n)\prod_i \pi_X(U_i) \quad (7)$$

$$= \sum_{U_1 \cdots U_n} p(X|U_1,\ldots,U_n)\prod_i p(U_i|e_{U_iX}^+) \quad (8)$$

Z is a normalization constant.
All equations follow from the conditional independencies in the graph.